GIMMAP

Geodata Interactive Management Map Analysis and Production

by

Charles G. Ross

B.A., Mathematics, The University of Kansas, 1974

Submitted to the Department of Computer Science and the Faculty of the Graduate School of The University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

Professor in Charge

Committee Member

For the Department

Kansas Geological Survey
Open-file Report

Disclaimer

ABSTRACT

A Geographic Data Processing (GDP) system has been designed and implemented on a minicomputer to fulfill the cartographic needs of the Kansas Geological Survey.

The Geodata Interactive Management Map Analysis and Production (GIMMAP) system provides the basic GDP functions of (1) accurate capture of map data through digitization; (2) creation and maintenance of a cartographic database which accurately models the input document; (3) interactive graphical support for display, correction, and addition of map data; and (4) user-specified retrieval of map features for map production, data transfer, and data archival.

These GDP functions are provided by ten FORTRAN programs and three utility libraries which have been designed and implemented with strict reliance on the principles of modular programming, physical and logical data segmentation, and extensive use of random-access disk files.

The GIMMAP system has reached a state of development which can support the basic functions of a GDP system in a restrictive minicomputer environment. The Geologic Map, Lawrence West, Kansas has been published, and GIMMAP is now serving to create and maintain a statewide cartographic database for production of geologic and other maps of any area of the State.

ACKNOWLEDGMENTS

I wish to express a great deal of gratitude to Dr. Tho Trang Cao of the Bureau de Recherches Geologiques et Minieres who introduced me to Computer Graphics and Cartography, which are the cornerstones of GIMMAP. His research and vision led to the creation and development of this thesis.

My thanks go also to William Hambleton, Director, and John C. Davis, Chief of the Geologic Research Section, at the Kansas Geological Survey for their enduring patience and support.

I would like to express a special note of thanks to Joe Brentano who has assisted me for the past three years. He has willingly provided all forms of support from digitizing maps to designing and implementing programs.

I want to thank Jo Anne DeGraffenreid for typing and correcting this manuscript; Chris Roche, Mahvash Sharifi, and Gina Aspiazu for their patient production of numerous plots; Joan Jaeger for production of illustrations; Julie Jaeger for additional typing; and Jim Deputy for software input.

Finally, I thank my wife, Chris, without whom this thesis could never have been completed.

C.G.R.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

SECTION I

INTRODUCTION

I.1 Thesis

A number of Geographic Data Processing (GDP) systems have been developed on large, "mainframe" computing systems. These mainframe computers often lie beyond the human and fiscal resources of small or intermediate sized institutions, which are able to support only the smaller, less costly "minicomputer" systems. Previous attempts to develop GDP systems on minicomputers have resulted in hybrid systems which rely on a mainframe computer to perform at least part of the processing (Steiner, 1979).

The purpose of this thesis is to demonstrate that a Geographic Data Processing system need not rely at all on a mainframe computer, but instead, may be implemented in the restrictive environment of a minicomputer system. Such a GDP system may be designed and implemented to effectively and efficiently support the basic GDP functions of (1) accurate data capture, (2) generation and support of a cartographic database, (3) graphical display of data, (4) interactive graphical editing (correction), (5) attribute and graphic data update (addition), (6) selective retrieval based on location and attribute values, (7) production of intermediate plots for edit-completion testing, (8) production of high-quality, color-separated scribecoats for multi-color map printing, and (9) retrieval of "clean" data for transfer or archival operations.

The Geodata Interactive Management Map Analysis and Production (GIMMAP) system has been designed and implemented at the Kansas Geological Survey to provide these GDP capabilities. The GIMMAP system has been implemented on both a Data General Nova and a Data General Eclipse minicomputer system. All GIMMAP program modules are written in the

FORTRAN IV language, and each module requires less than the maximum available (32K words) memory. The GIMMAP system has been thoroughly tested in a production environment, and is being used to construct a statewide database to fulfill the cartographic needs of the Kansas Geological Survey.

This presentation of the GIMMAP system consists of six sections. The first section states the thesis, introduces basic GDP terminology, reviews previous research from the literature, outlines some existing GDP systems, and provides a review of the development of GIMMAP at the Kansas Geological Survey.

Section II presents an overview of the GIMMAP system. Chapter II.2 introduces the terminology required for a general understanding of the system. Chapter II.3 details the operation of data capture (digitizing). Chapter II.4 presents an overview of the software and an operational flowchart describing the proper application of the programs. The structure and content of the GIMMAP database is described in Chapter II.5. The basic concepts, motivation, and operational procedures involved in interactive graphical editing are presented in Chapter II.6. Finally, a brief description of hardware requirements is provided in Chapter II.7.

Section III provides a catalogue of the programs in the GIMMAP system. Each program is described in one of 11 separate chapters, presented in alphabetic order by program name. The information in each chapter is divided into sections which (1) describe the purpose or functions of the program, (2) define local terminology relevant to further descriptions of the program, (3) detail the program operation and the program/user interaction, (4) describe and explain termination and error reports produced by the program, and (5) profile programming considerations including database access and subroutine calls.

Section IV introduces the Random File Management (RAFMAN) and Graphics Management (GRFMAN) subsystems which are incorporated in GIMMAP as utility libraries. All routines in these libraries are functionally defined, and the calling sequences and associated parameters are fully described. The RAFMAN chapter (IV.2) details the random-access management strategy employed by GIMMAP, and describes the nature and function of the RAFMAN routines. An introduction and categorization of the several types of graphics available in the GRFMAN library is provided in Chapter IV.3.

Section V describes features local to the implementation of GIMMAP at the Kansas Geological Survey. This section is presented to suggest one alternative in dealing with implementation strategies for the GIMMAP system. Included in this section are chapters on file naming schemes, the use of U.S. Geological Survey quadrangle maps as input documents, map projection and unprojection techniques, and a training manual for digitizing operators. The treatment of these concepts at the Kansas Geological Survey may serve to direct implementation efforts at other installations.

Finally, Section VI presents a review of the goals and achievements of the GIMMAP system. Conclusions are drawn from the experiences at the Kansas Geological Survey, and further research is outlined.

I.2 Geographic Data Processing

Many terms and definitions may be found in the literature to describe the application of information processing techniques to geographic data. Among these terms are "Computer Cartography," "Automated Cartography," "Map Data Processing," "Computer-Assisted Cartography," and "Geographic Data Processing." The term "Geographic Data Processing" (GDP) shall be adopted here to review several sources of general information in the GDP field.

A survey of specific GDP systems and associated users may be found in Teicholz and Dorfman (1976), in which computer cartography is defined to be "...the application of computers towards the design and production of maps."

Some desirable characteristics of GDP systems and brief descriptions of a sample of less known systems created for various sponsoring institutions is found in Brooks and Pease (1978). In this text, the authors define a geographic information system as a "computerized system designed to store, process, and analyze spatial data."

A most impressive work is the comprehensive review of the broad spectrum of GDP, edited by Tomlinson (1972). This two-volume set, prepared for the International Geographic Union (IGU), collected the thoughts and works of most leaders in the field at that time, and covered software, hardware, and specific systems up to 1972.

Tomlinson derives a definition for a geographical information system by combining a previous definition of an information system (Dueker and Horton, 1971) with the provision that "the data be referenced in a manner which will allow retrieval, analysis and display [based] on spatial criteria." Later, he adds "...it is generally agreed that a geographical information system contains data with location identifiers, that these data are manipulated and retrieved on geographical criteria and that the output generally takes the form of graphical presentation."

The common element in these and other definitions is the spatial nature of the data for which GDP systems are designed. It is also evident that specific application goals and limitations of hardware or other resources will produce design and implementation problems which may vary greatly among specific systems. In this vein, several reviews of GDP systems have been restricted to specific application areas. The application of geographic information systems to land planning is reviewed by Calkins and Tomlinson (1977) for the IGU Commission on Geographic Data Sensing and Processing. Applications to resource analysis were compiled by Davis and Lopez (1978) from the COGEODATA International Conference on Computer Mapping held in Mexico City.

Two other works concerned with the mainstream of general GDP applications are the excellent treatment of the inherent difficulties in geographic information systems software by Marble and others (1978), and a concise survey of GDP literature by Nagy and Wagle (1979a). The latter carefully unifies the contributions of the diverse disciplines which the authors view as being relevant to GDP, outlines the scope and functions of GDP systems in general, and discusses the implementation of these functions in the context of ten well-known GDP systems.

Current Systems

There exist many GDP systems which have been fully implemented and have been proven effective after long periods of testing. Many of these successful systems have been developed as general purpose cartographic systems, but some were restricted for specific applications only.

Most notable is a collection of programs developed since 1971 and currently available under the name ODYSSEY at the Laboratory for Computer Graphics at Harvard University. The basis of ODYSSEY is the well-known SYMAP program, designed by H. T. Fisher. The basic function of SYMAP is to produce contour and conformant maps on a line printer using various gridding techniques. Additional programs in the ODYSSEY package provide capabilities for the entry of digitized map data, production of plotted maps, interactive editing of map files, and production of 3-D views of map data. Applications of SYMAP are discussed in Liebenberg (1976), and an overview of the ODYSSEY system is presented in Dutton (1977).

An equally successful system is the Canada Geographic Information System (CGIS). The main function of CGIS is to input, store and display coverage maps. But, as suggested by Nagy and Wagle (1979a), its greatest importance may well lie in its contributions to the field of geographic data processing. The ability to "overlay two or more coverages for a region and to compute the areas of simple or compound coverages," and the use of coordinate chains to describe region boundaries are but two examples. CGIS is also known for its use of state-of-the-art hardware (e.g., optical scanners for input data) and consequently its large budget (Tomlinson and others, 1976). Although some aspects of the CGIS system have been negatively criticized in a paper by Boyle (1979) at the University of Saskatchewan, Nagy and Wagle (1979a) report that CGIS has been considered for adoption by the U.S. Geological Survey.

Several other general-purpose GDP systems are worthy of mention. NORMAP, developed in Sweden by Nordbeck and Rystedt, is similar in function to SYMAP, but is statistically more sophisticated. A detailed description of NORMAP, including some case studies, and discussions of general cartographic applications may be found in Nordbeck and Rystedt (1972).

The Experimental Cartographic System, created at the Rome Air Development Center (Diello, 1972), and SACARTS, developed at the Defense Mapping Agency Topographic Center (Struck, 1975) both rely on photographic input data and sophisticated hardware. And finally, GIRAS, a system developed by Mitchell and others (1977) at the U.S. Geological Survey, is designed to process land use and land cover data. GIRAS is less well known, but bears perhaps the strongest resemblance to the GIMMAP system.

Recent Developments

There are two traditional problems in most GDP systems which have led to very significant hardware and software developments in the past several years. These recent developments have been forged by the fusion of specialists from several disciplines to create tools which appear to eliminate major bottlenecks and seemingly revolutionize geographic data processing.

The first problem, discussed by Rhind (1974), is data capture, the conversion of cartographic data from maps or other sources to digital (computer-readable) form. A second major problem lies with the interface between sophisticated database management systems and relatively untrained managers, geographers, geologists, and the like. The former problem results primarily from the volume of the data, and suggests a lack of adequate hardware capabilites. The latter problem arises, perhaps, from the fact that the users generally provide input to the design, but not the implementation of the system.

The data capture problem has led to the development of raster scan digitizing equipment which "automatically" converts graphical data to digital form in a matter of minutes for the average map. Initially, such systems were thought to be impractical partly because of the initial expense of the hardware, and partly because of the overhead required to perform various editing and conversion functions. (The conversion is necessary since scan digitizers produce a raster representation, while systems employing these devices were geared to the vector representation.) But recent hardware and software advances, outlined by Boyle (1979), have reduced these costs to acceptable levels which may be preferable when compared to traditional digitizing costs.

Perhaps the most promising result of raster-oriented scan digitizers has been the influx of raster processing hardware and software previously developed in remote sensing and other fields. This natural progression has spawned numerous algorithms for vector/raster conversions (Rosenfeld, 1979; Mueller and others, date unknown; and Nagy and Wagle, 1979b) and various data structure designs (Rosenfeld and Samet, 1980, Cederberg, 1980). Inevitably, vector-raster hybrid systems have emerged (Steiner, 1979), and some have called for the exclusive use of raster graphics in cartography (Peuquet, 1979).

The second problem, mentioned above, concerns the use of sophisticated database management systems by relatively unsophisticated users. The emerging solution to this problem is originally rooted in a non-GDP setting at IBM where Codd (1970) introduced the concept of a "relational" database. This concept is strictly a logical view of data and operations on data by means of the well-known relational calculus.

The relational database concept has been very successful in terms of the number of (non-GDP) systems in which it has been used. This may be attributed to the fact that this model of data achieves a major objective of "data independence" as introduced by Codd and later described by Date (1972). Furthermore, the relational view of data provides "a rich set of commands" (Ullman, 1980), and allows the user to view the data as "a collection of named tables" (Nagy and Wagle, 1979a).

An early application of the relational model to GDP systems was the addition of software to the existing INGRESS relational database system to create a limited "geographic information retrieval and display" system called GEO-QUEL, described by Berman and Stonebraker (1976). Another GDP system called GADS is based on the relational model, and was developed at IBM, where Codd first introduced the relational concept. Finally, a "spatial data structure" has been introduced by Haralick and Shapiro (1980) to impose the relational view on both graphical and attribute data in a GDP system.

I.3 The Development of GIMMAP

The original kernel of the GIMMAP system was developed by Tho Trang Cao, a mathematician at the Bureau de Recherches Geologiques et Minieres (BRGM) in Orleans, France. From late 1974 through 1976, Dr. Cao developed the topological structures and associated generating algorithms to provide computer assistance for the production of two experimental geologic maps of areas in France. The result of Dr. Cao's efforts was the production of scribes containing the geologic contact lines which were used to prepare color-separation materials for printing.

In the summer of 1977, Dr. Cao became the Visiting Research Scientist at the Kansas Geological Survey, bringing with him the programs he had developed at the BRGM. He also brought a wealth of knowledge of the cartographic process and computer graphics which he had gained in his experiments. During the academic year 1977-78, Dr. Cao and the author collaborated to implement the basic system (then called GIMMS) incorporating the digitizing process, the basic file structures, the editing facility and the generation algorithms (Cao, 1978) which Dr. Cao had developed.

The primary concern of this initial development was that the system was to be implemented on the K.G.S. Data General Nova 1220 minicomputer with 32K, 16-bit words. It was apparent that the system must be developed as a group of independent program modules, and that the data must be segmented logically and physically in random-access disk files.

Dr. Cao defined the basic unit of data as the county-patch (Chap. V.2) and all map data were segmented according to the unique areas defined by these units. Further segmentation by feature type (e.g., transportation, hydrology, geology) was defined by use of the feature code (Chap. II.2) to provide layers of data for each county-patch. This additional level of data segmentation may be required when the density of the data is very high.

The author designed and implemented the Random File Management (RAFMAN) subsystem described in Chapter IV.2. The purpose of the RAFMAN subsystem is to support all basic functions of random-access disk files within the framework provided by the operating system. This random-access approach made possible immediate and direct deletion and update functions required for interactive editing (Chap. II.6) and reentrance of the generation programs (Chap. III.6, III.7).

Dr. Cao designed the basic procedures and syntax (Chapter II.3) for digitizing on the Bendix digitizing table at the Kansas Geological Survey. The primary advantages of Dr. Cao's digitizing system were (1) a uniform approach to digitizing all data; (2) elimination of the need for entering header information, line or node numbers, and left-right codes; and (3) elimination of the need to digitize zone identification marks and multiple digitization of zone boundaries.

These digitizing procedures were later modified by the author to simplify the tedious task of digitizing. The author formalized the syntax as a transition matrix within an interactive syntax-editing program, to perform a task which had previously been performed with the system editor. This syntax editor was later transformed into a non-interactive, "automatic" syntax editor called SYNEDT, described in Chapter III.10. The algorithms for generation of the Point, Arc, and Node files, embodied in the original PARGEN (Chapter III.7) and NODGEN (Chapter III.6) programs, were implemented basically as they had been by

Cao, with the incorporation of the random-access approach for the Arc, Point, and Node files. Subsequent changes in the syntax by the author required modifications in the generation of these files.

One of the drawbacks in the initial PARGEN and NODGEN programs was the inability to add arcs and nodes once the base files were created. In this situation, the input files had to be concatenated, and the programs were re-run. The author modified both PARGEN and NODGEN to incorporate the options of initialization and re-entrance, eliminating the redundant generation of the Arc, Point, and Node files.

The author also created a paging system for the Point file, and removed the redundant storage of Arc endpoint coordinates in the Point file, since these coordinates were also stored in the arc records. The paging system required pointers to be added to the arc records to support immediate access of the interior points of arcs. The previous system at BRGM relied on large block access for point coordinates, requiring secondary searching and time-consuming disk I/O. The paging system reduced this overhead with immediate access and minimal disk I/O.

Dr. Cao had previously developed an editing program which provided the ability to display all data, window the viewing area, change arc feature codes, and delete arcs. This program was designed for the Tektronix 4014-1 graphics terminal, and utilized the Tektronix PLOT-1Ø graphics software. The availability of a similar Tektronix terminal and PLOT-1Ø software system at the Kansas Geological Survey induced Dr. Cao to introduce the author to the fundamental concepts and procedures of computer graphics. With this introduction, and the guidance provided by Dr. Cao and his initial program, an extended interactive graphical editing system was designed and implemented.

This program (then called EDITLOOK) incorporated the new base file structures, and new editing commands to change point locations and to join arcs by matching their nodes. This program also used the PLOT-1Ø graphics routines to perform all Tektronix graphical functions. Subsequent modifications by the author resulted in the addition of commands to support masking of arcs in the display by means of a background code, changing the position of nodes, creation of new nodes on existing arcs (arc splitting), automatic numbering of nodes for identification of features, and a new zoom windowing feature to allow tracking of features outside the display.

Additionally, it was decided that the methods used by the PLOT-1Ø software provided a display that was far too slow when operating on the larger data sets involved in actual maps. Thus, the author investigated the low-level functions involved in the display of data on the Tektronix, and developed a quick-plotting (Qplot) display file to be accessed for normal plotting of data at the original display scale. This tripled the speed of the display, but could not be used for windowing of the data since it relied on absolute screen coordinates. The eventual solution to this problem came later, when the PLOT-1Ø software was totally abandoned.

In the last months of Dr. Cao's stay at the Kansas Geological Survey, a student programmer (Joe Brentano) was hired to aid in the digitization and editing of an experimental map, and to provide programming support for various utility projects.

During this period, the author implemented Dr. Cao's left-turn procedure (Cao, 1978) in the context of the KGS system (ZONGEN, Chapter III.12) and designed and implemented the basic program (ZONEDT, Chapter

III.11) used to interactively edit and update zone information. Additionally, a retrieval program (PLTGEN, Chapter III.8) was designed and implemented to provide output used to drive the XYNETICS plotting software for the production of paper plots and scribecoats.

The ZONEDT program updates various zone attributes and relationships as introduced by Dr. Cao. These features then allow for the separation of zone boundaries according to the zone color codes which represent the basic attribute values of the zone. The retrieval process may then create individual files to produce the color-separated scribecoats needed for the map printing process.

By the time Dr. Cao returned to France in 1978, the processing of data for the experimental geologic map of the Lawrence East quadrangle was complete. The map was subsequently produced as a color-proof, but was not published. At this time, work was begun on a second, more complex geologic map of the Lawrence West quadrangle.

The improvement and extension of nearly all program modules continued for the next two years, principally by the efforts of this author, assisted by Joe Brentano. The most significant developments were additions and modifications of the graphical editing program and the creation of the GRFMAN subsystem (Chapter IV.3) to replace the PLOT-1Ø software for all graphics operations. The editing program became the GRFEDT program, and the results of the extensions and modifications of this program are reflected in the materials in Chapters II.6 and III.5. Both the GRFEDT and ZONEDT programs have been modified to access the GRFMAN utility library for all graphics functions.

The PLTGEN program, used for selective retrieval of graphical data from the database, has been enhanced with map titling, map border creation, node numbering, and feature code selection capabilities. These new features support production of edit-completion testing plots, as well as production of high-quality scribecoats. Several procedural modifications to prevent multiple-plotting of arcs, and to minimize pen movement have also been added.

In 1979, the author visited Dr. Cao at the BRGM in Orleans, France, for two months. During that period, much of the material contained in Section III was written. Additionally, a new algorithm for generation of nodes from the Arc file was designed and implemented. This new algorithm allows for the creation of up to 3000 nodes from a set of 3000 arcs, while the original NODGEN program was limited to 400 nodes from 500 arcs. The new NODGEN program (Chapter III.6) has proven to be more useful in terms of the actual number of arcs and nodes in maps, and operates faster than the original program.

In late 1979, the digitizing and editing for the geologic map of Lawrence West were completed. The scribecoats were produced and a color-proof was prepared for presentation at the Fourth International Symposium on Computer Assisted Cartography (AUTO-CARTO IV) in Reston, Virginia. The Lawrence West geology map was published in 1980 as Kansas Geological Survey Map M-14 (Ross and McClain, 1980), and the AUTO-CARTO IV presentation was published in the Proceedings (Ross, 1980).

In 1980, the entire GIMMAP system was converted for operation on the new Data General Eclipse minicomputer at the Kansas Geological Survey. This conversion process required only minor modifications, primarily in the input and output operations, and spawned a set of (system-specific) utility functions to improve the operator interface.

Also in 1980, the Kansas Geological Survey undertook to apply the

GIMMAP system to the creation and support of a Kansas database (KDB) of cartographic information. To support this effort, a manual for digitizing operators was compiled (Chapter V.5), and a training procedure for digitizing and editing using GIMMAP was instituted. The construction of the Kansas Database has progressed under GIMMAP, and currently, nearly one-fourth of the State's USGS 7.5′ quadrangle maps have been entered.

The most recent development has been the incorporation of map projection and unprojection software (Chapter V.4). This software provides rectification of map boundaries with the true projected size and shape dictated by the mathematical model. The rectification is necessary for inter-map editing and concatenations of quadrangle maps to form larger scale maps of larger areas such as counties. The projection software has been applied to produce a base map (Ross and Brentano, 1980) of Kansas Groundwater Management District No. 4 from the approximately 120 quadrangle maps. This map is being used by the Geohydrology Section of the Kansas Geological Survey to serve as a base for future geohydrologic research maps.

The current and future directions of research and development of the GIMMAP system at the Kansas Geological Survey will be discussed in Section VI.

SECTION II

SYSTEM OVERVIEW

II.1 Introduction

This section presents a comprehensive overview of the GIMMAP system. This overview is intended to be general in nature, with the specific details of the software to be introduced in subsequent sections. Factors which may vary between installations will be treated in general terms. These factors are discussed in the context of the Kansas Geological Survey implementation of GIMMAP in Section V.

The section begins with the GIMMAP view of map data and an introduction to the terminology used in subsequent chapters. Next, the subject of map data input by digitizing is introduced, and the GIMMAP procedures concerning data input are described in detail. An overview of geographic data processing functions and their realization in specific program modules leads to a detailed presentation of the operational sequence and general functions of GIMMAP program modules. (The program modules are presented in greater detail in Section III.) Next, the database structure, file contents and interconnections are described, and the relationship between program modules and database files is outlined. An introduction to the motivation and specific methods involved in map data editing and updating is given as preparation for the more detailed operational guide presented in Section III. Finally, a brief description of hardware requirements details the minimum configuration and describes the configuration on which GIMMAP has been designed.

II.2 GIMMAP Terminology

This chapter introduces and defines the terms which are necessary to discuss the general functions of the GIMMAP system. Specific terminology required in individual functions such as digitizing, interactive editing, and base file generation will be introduced as required in subsequent chapters.

A map consists of graphical features which may be classified as points, lines, and areas. Wells, small buildings, towns, quarries, land grid points and other isolated objects may be represented by point symbols. Highways, contour lines, rivers, railroads, and areal boundaries are represented by lines. Areas, denoted by color, shading and/or actual linear boundaries, may represent vegetation, geologic formations, hydrologic features or other conformant information.

The capture of map or graphical data is called digitization. Various types of digitizing devices allow an operator to convert points on a map into computer-readable Cartesian coordinate values based on a pre-defined origin and set of axes. Thus, a single point is represented in the computer as a pair of x-y coordinates, and a line is represented by a sequence of such points. This representation is compatible with output devices called plotters which may be instructed to draw lines from one point to another.

Linear features may intersect the map boundary or each other, or may end at any point within the map boundary. Such intersections or terminations are referred to as nodes, and are represented in GIMMAP as a single pair of x-y coordinates. Nodes divide linear features into smaller segments called arcs, each consisting of two endpoints (the nodes) and a (possibly empty) string of intermediate or interior points. Point features are represented in GIMMAP as special arcs called isolated points or simply points. These arcs have no interior points, and both endpoints are the same node. They may be considered to be "collapsed" arcs.

The number of arcs which intersect at a single node is called the degree of the node. While this number may, in theory, be arbitrarily large, actual map data suggests that node degrees are usually less than seven. This fact, coupled with the programming goals of rapid access and limited waste, suggest a maximum node degree of six arcs intersecting at one node. This restriction is made less burdensome in GIMMAP by restricting nodes to be the intersections of arcs of common feature types. For example, the intersection of a river and a contour line would not be considered to be a node.

Areal map features are called zones and are represented by ordered sequences of (non-isolated point) arcs. The sequence of arcs which represents the closed boundary of a zone is called the zone boundary, and the arcs comprising the boundary are called border arcs. It should be noted that these arcs are digitized in the same manner as any other linear features. The construction of the actual boundaries is performed by the software, not by the digitizing operator.

The interior of a zone is the region located to the left of the zone boundary when this boundary is described in a counter-clockwise fashion. To allow graphical identification of a zone, a point within the interior is artificially selected as the zone mark. This selection is done interactively after the zone boundaries have been compiled.

An island is a zone which lies completely within the interior of another zone. The outer zone is called a surrounding zone, and may contain any number of islands, any of which may also contain islands. Thus an island may also be a surrounding zone.

Although all zone border arcs have an associated feature code, each zone has a particular attribute which is identified by a zone color code. This code is set interactively when the identifying zone marks are chosen. At that time, it is also possible for an operator to select several points for the eventual placement of zone labels on the resulting map. These labels serve to identify by name the attribute specified in the color code.

In order to distinguish different types of features, a pre-defined set of feature codes is created. Each feature code consists of four digits, and each digit connotes some information about the type of feature. In the Kansas Geological Survey implementation, the first digit indicates the graphical type. A "1" (one) identifies the feature as an isolated point. Arcs which are not considered as zone boundaries have feature codes beginning with a "2" or "3", and zone border arc codes begin with a "4". Further digits represent various classifications and sub-classifications within a given graphical type. This scheme is exemplified by the set of feature codes used in the Kansas Geological Survey implementation shown in the Appendices.

Map data which has been digitized is converted into disk storage files, collectively called base files. This set of random-access files consists of the Arc file, Node file, Point file, and a display file called the Qplot (Quick plot) file. The conversion or creation process is called base file generation. When these files are to be generated, they are first initialized to indicate that they contain no data. After the initial generation, additional data may be subsequently appended through reentrant generation which preserves all preexisting data.

Once the base files have been generated, digitizing errors may be corrected, and updated information may be incorporated through a process called interactive graphical editing. This process is performed with a graphic display terminal which is capable of drawing lines on the screen to display the map data in visual form. Such a terminal may also allow an operator to graphically specify features of the map by using a cross-hair cursor to identify (to the computer) any point on the display screen.

Interactive editing in GIMMAP provides for the modification of arcs, nodes, points, and codes for the displayed map. To facilitate identification of features to be modified, numbers are assigned to all features during the generation of the base files. Thus, an operator may identify features by use of node numbers, which may be displayed on the screen by a simple command.

In interactive editing, the identification of arcs by node numbers representing the endpoints will be unique except when two or more arcs have the same two nodes as their endpoints. This phenomenon occurs frequently and is termed bridging. This may occur when a single arc is erroneously digitized twice. In this case, the two arcs are called duplicate arcs.

When the bridging phenomenon occurs, the digitizing operator should select an interior point on one of the arcs, to be used as an artificial node. This point is then digitized as a node rather than an interior point, and will result in a unique pair of nodes for all arcs. If more than two arcs bridge a pair of nodes, all but one must be split by artificial nodes. This operation may also be performed during the editing process.

When the base file editing is complete, a secondary set of base files may be generated if the map data contains areal features (zones). This generation results in the creation of the Zone file and the Border file. Interactive graphical zone editing may then be employed for the addition and updating of zone marks and colors. Zone editing allows for functions paralleling those of base file editing, but also allows for the addition of label information, and thus the generation and updating of the Label file.

The result of the generation of the extended base files is the ability to perform selective retrieval of the map data for the preparation of plotted maps or scribe-coat material for color-separation and eventual map production. In this process, map data may be selected by feature numbers or by code ranges specifying groups of feature codes to be selected. Zones may be selected by zone numbers or by zone colors. Headers (consisting of textual information such as map titles), and fiducial marks (used for registration of overlays), may be added during this process. Similar retrieval capabilities are available to produce output in the same form as the original digitized input data. This output may then be used to transfer map data or to store data in an archive.

II.3 Data Input

The object of digitizing is to create a file of data which comprehensively describes the input document (map) and which is suitable for processing by computer. A set of digitizing rules, called the syntax, describes the order of operations, content of data records, and format of the output file. The GIMMAP syntax has been developed to impose the GIMMAP view of map data upon the process of capturing data by digitization. The end result of the application of the digitizing syntax is a digitized data file.

Digitized Data File

The digitized data file consists of three parts. The first part is the map header, which is the name of the map and any other desired textual information. The second part is a set of control points which are the coordinates of the corners of the map (or enclosing rectangle). The final part of the file contains the coordinates of feature information such as points and arcs. The digitizing procedure consists of entering the map header at the terminal, digitizing the control points (counter-clockwise from the origin), and digitizing the features of the map.

Basic Feature Digitizing

Basic feature digitizing is illustrated in Figure II.3.1, in which a map containing two rivers and a road is shown as a group of nodes, arcs, and isolated points. To the digitizing operator, the map consists of two arcs of the same code (1-2 and 3-4), and a third arc (5-6) of another code. In this view, the circles are nodes, the squares are

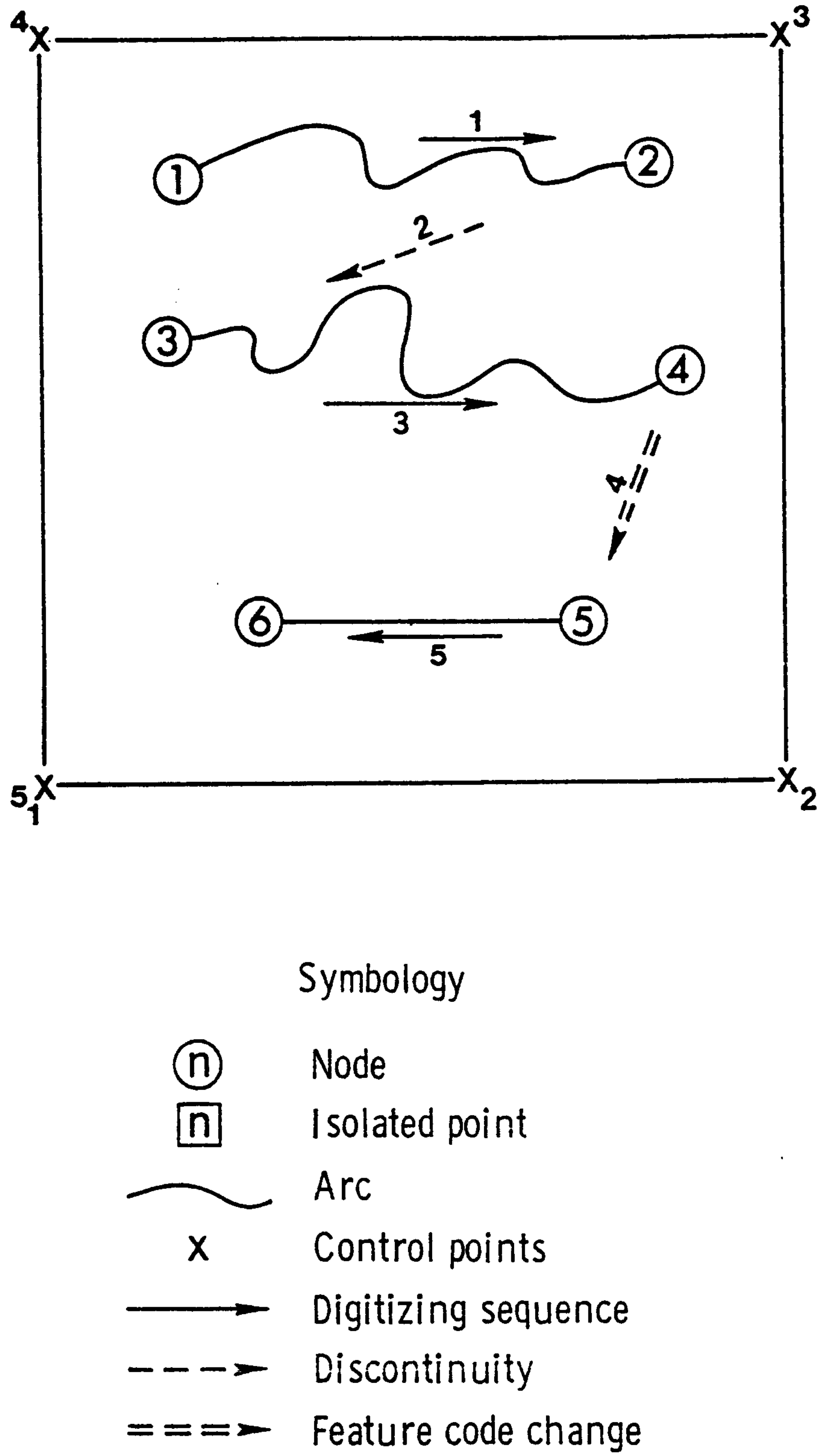


Figure II.3.1. Basic Feature Digitizing

isolated points (Figure II.3.2), and the solid lines connecting nodes are the arcs. The control points defining the map boundary are marked by X, and are numbered according to the order in which they are digitized.

The digitization of points on the map occurs by placing the digitizing cursor such that the crosshairs of the cursor intersect at the point to be digitized. A single push of the cursor button transmits the x and y coordinates of the point (relative to the selected origin and axes) to the computer. To digitize an arc, the operator must view the (possibly) curved line as a sequence of straight line segments, and digitize each endpoint in order. Each of these interior points is digitized by pushing the cursor button one time.

To digitize nodes, the operator pushes the cursor button twice so that two identical sets of coordinates are transmitted. Thus, in the digitized data file, any pair of consecutive identical points (within a small tolerance) indicates the presence of a node rather than an interior point. An exception is made for isolated points. Since all isolated points are nodes, there may be no interior points for these arcs. Isolated point feature codes act as a signal that each point is a node, and the operator must digitize these points only one time each.

To completely digitize our sample map, the operator first enters the map header (containing the name and perhaps other information) at the keyboard, and then digitizes the control points (one time each) in a counter-clockwise fashion, beginning and ending at the lower-left corner. Next, the feature code for rivers is entered from the keyboard as a pair of x-y coordinates, called the arc header, with the x-coordinate equal to the feature code. This feature code will be applied

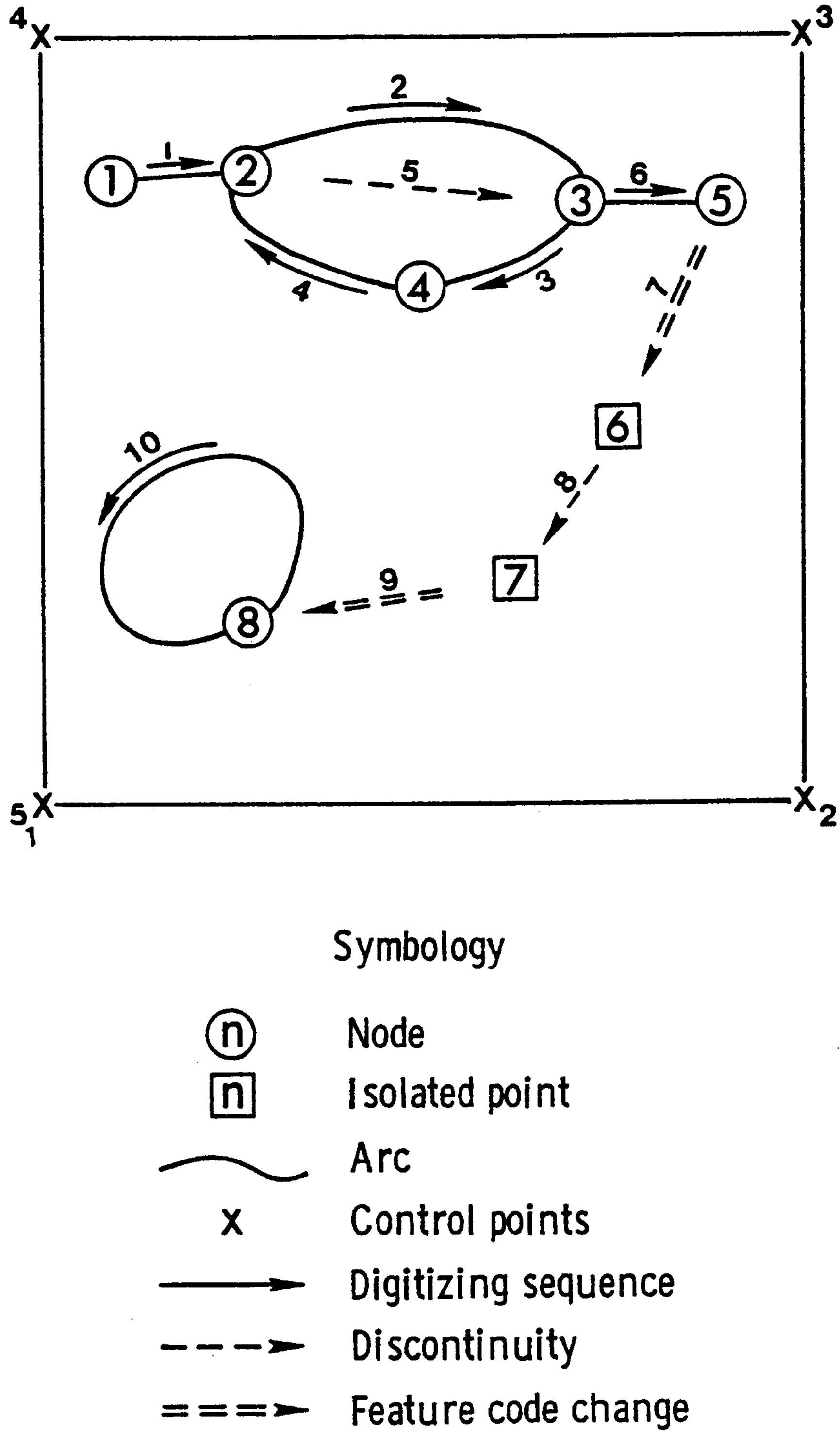


Figure II.3.2. Special Digitizing Procedures to subsequent arcs until the next arc header is entered, thus eliminating the need to enter an arc header for each feature.

After the arc header is entered, the first node of the arc to be digitized, called the start node, is digitized. In the example, the start node for the first arc is node 1. The operator digitizes this node twice, digitizes each of the interior points once, and digitizes the end node (node 2) twice.

The next feature to be digitized will be the second river arc from node 3 to node 4. Since the feature code for rivers is still in effect, the operator does not need to enter a new arc header. Furthermore, this arc does not begin at node 2 where the cursor is located following the digitization of the first arc. This means that the cursor must be moved to the next starting node without digitizing any points. This is a discontinuity, which must be indicated in the digitized data file. Hence, the operator enters a discontinuity flag (represented by the single, dashed arrow) at the keyboard. This flag consists of a pair of coordinates with negative x and y values (usually -1.0, -1.0). Since the map has been situated so that all points of the map have positive coordinates, the system is able to distinguish flags from other points.

The operator then digitizes the start node twice (node 3), the interior points, and the end node twice (node 4) for the second river arc. No arc header is required because the feature code is the same as for the first river arc.

Next, the operator will digitize the road arc from node 5 to node 6. In this situation, both a discontinuity and a change of feature code must be indicated. The operator enters a header flag (represented by the double, dashed arrow), consisting of two pairs of negative coordinates. This is equivalent to entering two consecutive discontinuity flags. Entered at the keyboard, the header flag indicates that the next pair of coordinates will be a header, containing a feature code as the x-coordinate. The header flag also serves to indicate that the previous node is an end node.

The operator then enters the arc header containing the feature code for roads as the x-coordinate. For this feature (5-6), let us assume that the road is a short, straight arc requiring no interior points for accurate representation. In this case, the operator digitizes the start node (5) twice and then immediately digitizes the end node (6) twice. Since this is the end of the data, a header flag is entered to signal that digitizing has been completed, and to cause the previous node (6) to be treated as an end node.

It should be noted that the labels "start" and "end" are used only for discussion of the digitizing process, and are entirely dependent upon the sequence in which the map is digitized. As will be shown in the next section, a node may be treated as a start node at one time, and as an end node at another time. In this example, nodes 1, 3, and 5 were treated as start nodes, and nodes 2, 4, and 6 as end nodes. Start nodes are always preceded by a discontinuity flag or by an arc header, and end nodes are followed by discontinuity or header flags.

Shortcuts and Special Cases

In Figure II.3.2, several special features of the digitizing syntax are illustrated. Let us assume that the map header and control points have been entered in the manner described above. The procedure for digitizing this example map is described below.

The arc header is entered for the arcs connecting nodes 1, 2, 3, 4, and 5, assuming these arcs all have the same feature code. The operator digitizes (start) node 1 twice and the interior points to node 2 (one time each). Since there is no discontinuity at node 2, the operator may continue digitizing to node 3 without entering a discontinuity flag, provided that node 2 is digitized twice. Node 2 is called a continue node since it acts as an end node for the first arc and a start node for the next arc. The operator digitizes node 2 twice, and continues digitizing the interior points to node 3 without entering a flag. The same procedure is repeated at node 3 (also a continue node) in digitizing the next arc to node 4.

Notice that if node 4 did not exist, nodes 2 and 3 would be connected to each other by two distinct arcs. This is an example of bridging, and if allowed to remain, ambiguity would arise in distinguishing the two arcs by their two node numbers. The bridging problem is avoided by arbitrarily adding an artificial node (node 4) to one of the arcs.

The operator digitizes node 4 twice and continues to node 2, which in this case is an end node. (Notice again that the terms start, end, and continue are procedural and relative; they are not defined by the graphic structures.) The operator enters a discontinuity flag after digitizing node 4 twice, then digitizes node 3 twice and the interior points (if any) to node 5 which is also digitized twice.

The next feature to be digitized (according to the sequence selected by the operator) is a Point (Isolated Point). The operator enters a header flag and a point header at the keyboard. Since the feature code in the point header indicates that subsequent points are nodes, the operator digitizes point 6 only one time. The next feature, point 7, is also an isolated point, and by default, there is a discontinuity between the two points. However, the point header serves as an indication of this situation, and the operator does not enter a discontinuity flag between the two points. It is possible that the feature codes for the two points may be different (but not in this example). If that were the case, a header flag and point header containing the new feature code would be entered following point 6. In this case, point 7 is digitized once following point 6.

The operator then enters a header flag and an arc header before proceeding to node 8. This arc, starting and ending at node 8, is an island. Since there is no node defined by arc intersection, the operator has selected an artificial node to serve as both the start and end node of the arc. Node 8 is digitized twice, the interior points are digitized, and node 8 is digitized (twice) a second time. It does not matter which direction the interior points are digitized, only that the same point is used for the start and end nodes. Since this is the end of the data, a header flag is entered to terminate the file.

This example has demonstrated some shortcuts and special features of the syntax. It should be noted that this map may be digitized (according to the syntax) in several different sequences resulting in different file structures, each of which would convey the same graphical information. As digitized, nodes 1, 3, and 8 were used as start nodes, nodes 2, 3, and 4 as continue nodes, and nodes 2, 5, and 8 as end nodes. Notice that nodes 2, 3, and 8 were used in different roles at different times.

The output data for the second example is shown in Figure II.3.3 and ten basic rules of the digitizing syntax are given below.

Digitizing Rules

1. The map header containing the map title and other textual information is entered first.

2. The four corners of the map are digitized as control points counter-clockwise, beginning and ending in the lower-left corner.

3. All headers (except the one following the control points) must be preceded by a header flag. The feature code (x-coordinate of the header) applies to all subsequent arcs until the next header.

4. All nodes are digitized twice.

5. Isolated points of the same code are digitized only one time, without interceding discontinuity flags.

6. When two unconnected arcs of the same code are to be digitized sequentially, the end node of the first and the start node of the second must be separated by a discontinuity flag.

7. Arcs may be digitized in either direction.

8. If two nodes are connected by two (or more) distinct arcs, artificial nodes should be selected to avoid the bridging problem.

9. Any point of an island may be selected as the start node. The same point is used as the end node.

10. All arcs, including zone boundary arcs between adjacent zones, are digitized once.

Map Header	Text Information
Control Points	X_1, Y_1 X_2, Y_2 X_3, Y_3 X_4, Y_4 X_5, Y_5
Arc Header	Feature Code, 0.
Start Node	X1, Y1 X1, Y1
Arc 1 Interior Points	
Continue Node	X2, Y2 X2, Y2
Arc 2 Interior Points	
Continue Node	X3, Y3 X3, Y3
Arc 3 Interior Points	
Continue Node	X4, Y4 X4, Y4
Arc 4 Interior Points	
End Node	X2, Y2 X2, Y2
Discontinuity Flag	-1. , -1.
(continued)	
Start Node	X3, Y3 X3, Y3
Arc 6 Interior Points	
End Node	X5, Y5 X5, Y5
Header Flag	-1. , -1. -1. , -1.
Point Header	Feature Code, 0.
Isolated Points	X6, Y6 X7, Y7
Header Flag	-1. , -1. -1. , -1.
Arc Header	Feature Code, 0.
Start Node	X8, Y8 X8, Y8
Arc 10 Interior Points	
End Node	X8, Y8 X8, Y8
End-of-File	-1. , -1. -1. , -1.

Figure II.3.3. Sample Digitized Data File

Preparation Guide

Careful preparation will help to reduce errors and increase the efficiency of the digitizing process. The following suggestions should be considered before the digitizing process begins.

1. A high-quality, physically undamaged map should be selected as the input document.

2. A complete list of all features (roads, rivers, etc.) which are to be digitized should be made available.

3. The contents of the map header must be determined.

4. A list of feature codes should be available for easy reference.

5. Criteria for deciding which arc intersections are to be digitized as nodes (usually, arcs of the same feature type) must be determined, and nodes may be lightly circled in colored pencil.

6. The map should be studied for islands and bridging phenomena, and artificial nodes should be marked.

7. Zones may be shaded lightly with colored pencil without obscuring other features.

8. The map should be firmly secured to the table with tape; wrinkling and stretching should be avoided.

9. The sequence (or route) should be carefully considered before any digitizing begins.

10. The digitizing guide (below) should be considered.

Digitizing Guide

The ten rules of digitizing (listed above) should be followed at all times. The following suggestions are supplemental hints to be considered before digitizing begins.

1. Digitize all isolated points first, all non-zone-boundary arcs second, and all zone-boundary arcs last.

2. Within one type (Point, Arc, Zone-boundary arc), digitize all features of the same code in one group (e.g., all rivers). The benefits from this approach are that only one header is entered for each group, fewer features will be accidentally overlooked, and the operator will better know what features remain to be digitized.

3. If there are many features of the same type and feature code, digitize them in an order similar to that of tiling a wall. This will help to reduce the number of features that are overlooked or that are mistakenly digitized twice.

4. Draw a light, but obvious, pencil mark through the arcs that have been digitized. This should be done at convenient times (e.g., after entering a flag) to minimize interruption of the operator's concentration.

5. Take careful note of the location of the most recent end node before entering the flag. (Usually, the cursor may be left there while entering the flag.)

6. Some digitizing systems may offer alternate methods for the entry of flags (a special cursor button, or digitizing a point with negative coordinates), and for the entry of header information. These methods should be employed if they are more convenient for the operator.

7. Interactive systems may allow the operator to correct digitizing errors if they are detected in time. Other systems provide no such facility, but detected errors may be minimized by the operator. When an error is detected during the digitizing of a feature, the operator may enter a header flag and an arc or point header and re-digitize the incorrect feature from the last correct point or from the beginning. The incorrect feature may be easily deleted at a later time.

Special Note

Many digitizing systems offer two modes of digitizing. These are point mode, in which a single point is transmitted at the push of a button, and line or stream mode, in which a continuous stream of points is transmitted while the button is depressed. Line mode may give the appearance of being faster, but experience suggests that this benefit must be weighed against superfluous data entry, weeding, editing, and re-digitization. While GIMMAP is designed primarily for point mode, line mode may be used, provided that the syntax rules are followed carefully.

II.4 Software Overview

Geographic data processing consists of the primary functions of data capture, editing, updating, and retrieval of cartographic data. To support these functions, GIMMAP software contains ten program modules and two supporting utility libraries. At the extremities of this processing, the actual digitizing and plotting functions are supported by non-GIMMAP software packages, usually provided with the hardware.

GIMMAP software is designed to operate in a minicomputer environment with a minimum of 32K 16-bit words. All software is written in standard FORTRAN IV, with the usual exception of input and output routines and the handling of alpha-numeric data. Interactive editing and updating facilities require the availability of random-access disk storage, and the access methods for this medium vary widely among different systems.

The functions inherent in geographic data processing suggest the basic sequence of operation shown in Figure II.4.1. In this primary sequence, each box describes a basic set of functions, and indicates the program module and relevant chapter which pertain to the function. Previous chapters in this section have described the processes of data preparation (step 1) and digitizing (step 2), and introduced the syntax, or rules governing the structure of the digitized data file. This file is the main concern of the SYNEDT program shown as step 3.

The syntax editing program, SYNEDT, relies on a transition matrix and a lexical analyzer to interpret and correct the digitized data according to the rules of the syntax. The transition matrix incorporates empirical, real-world knowledge to correct syntax errors and produce an output file which may contain graphical errors, but is

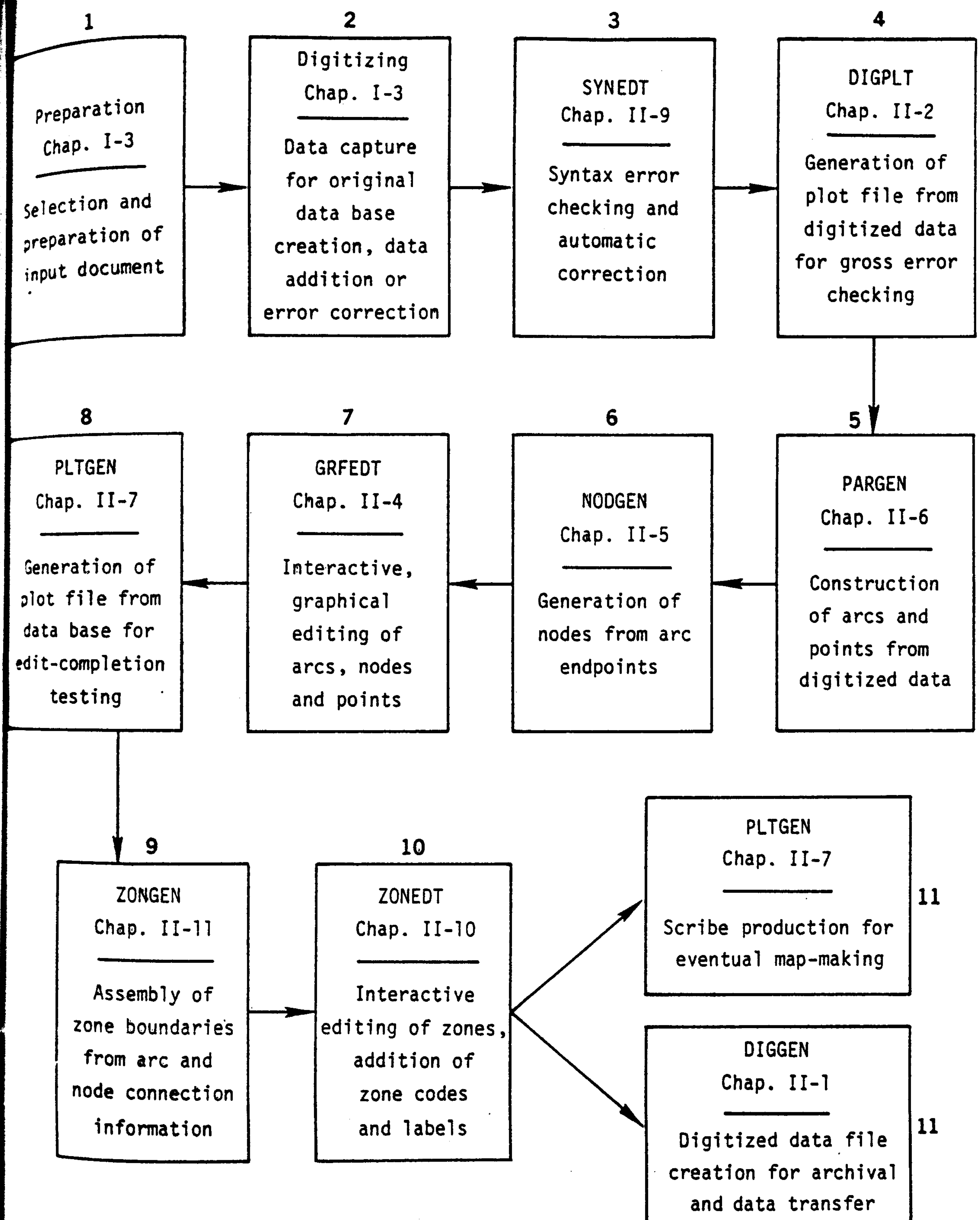


Figure II.4.1. Basic Sequence of Program Operation completely consistent with the syntax. The output file may then be plotted using the DIGPLT program (step 4) for an early check of gross graphical errors introduced during the digitizing process.

The syntactically clean data are then used by the PARGEN program (step 5) to construct arc and point records based on the rules of the syntax. Although not shown, the PARGEN program is also capable of rectifying the input data with the known map projection parameters for the given map. The major benefit of this operation is to force the shape and dimensions of the map boundary to be consistent with the mathematical model indicated by the map projection. This is necessary due to the graphical distortions inherent in paper documents.

The Node file is created from the Arc file by the NODGEN program to describe the topological network contained in the map. Shown in step 6, the NODGEN program resolves the unique locations and relationships of arc intersections to create node records. In this process, it is assumed that a digitizing operator may consistently digitize a given node within a fixed tolerance value at different times during the data capture phase. NODGEN determines the sets of arc endpoints which, indicated by their relative distances, represent a single node. For each such node, the average location is computed, and this location and arc-node connections are stored in a record in the Node file. Finally, each arc record is updated to contain the node numbers of its two endpoints.

At this point, the data are interactively edited using the GRFEDT program (step 7). This process employs a graphics display terminal to display the map data in graphical form. The actual display data are created by PARGEN, and reside in the Qplot (Quick-plot) file. The operator may add, delete, or change the graphical feature data using a simple command menu and graphical input to select various displayed features. This process is enhanced by the application of "background" codes to selectively display subsets of the data based on feature codes. In addition, GRFEDT provides two different commands to select "windowed" subsets of the data, based on their locations. The addition, deletion, and correction of data is immediate, and may be displayed in its new form following the update operation.

When the operator feels that the data editing is complete, an edit plot may be produced (step 8) using the PLTGEN program. The resulting file is then submitted to the plotting software, and a full-scale plot of the data with node numbers is produced. This plot may be overlaid on the original map to locate omissions and positional errors in the data. These errors may then be corrected with GRFEDT, or by additional digitization of the map. Additionally digitized data are entered via steps 1 through 6, using the reentrance options for the PARGEN and NODGEN programs.

Once the feature data are graphically correct, the ZONGEN program (step 9) is used to generate zone boundaries from an operator-selected subset of arcs. This process creates a zone record and a group of linked border records for each zone. In step 10, the ZONEDT program is used to perform several updates on the zone information. Using ZONEDT, the operator has the options of (1) selecting zone marks for display of zone identification numbers, (2) specifying zone color codes to identify the major attribute value of the zone, (3) selecting label marks and symbolic information for eventual placement of zone attribute labelling, (4) specifying island and surrounding zone relationships for proper zone boundary retrieval, and (5) linking of zones by common color codes for selective retrieval by these codes.

The final step in the basic sequence is the production of multiple plot files for scribe production of color separation materials, and subsequent multi-color map production. This selective retrieval of data using the PLTGEN program also allows for the introduction of map titles, map borders, and fiducial (registration) marks. The multiple plot files may also be used for one-time production of special-purpose maps.

As an optional final step, the DIGGEN program may be used to create a file in the same format as the syntactically clean, digitized data file. The same selective retrieval options available in PLTGEN are provided by the DIGGEN program to prepare a file to be used for archival or data transfer operations.

As indicated in Figure II.4.1, the program modules are more fully described in Section III. The two utility libraries used to support graphics management (GRFMAN), and random-access file management (RAFMAN), are described in Section IV. In addition to these modules, GIMMAP provides a program to perform consistency checks on the graphical network described by the node and arc files (GRFCHK), and a program to provide interactive display of the record contents of the base files (RAFPRN). While GRFCHK may provide information useful to the proper editing of the base file data, these programs primarily serve to locate and correct errors resulting from catastrophic events such as program error or an unexpected power-down of the system. These programs are also described more fully in Section III.

The interaction of the program modules with the database files is shown in Figure II.4.2. Program modules are shown as rectangles, and

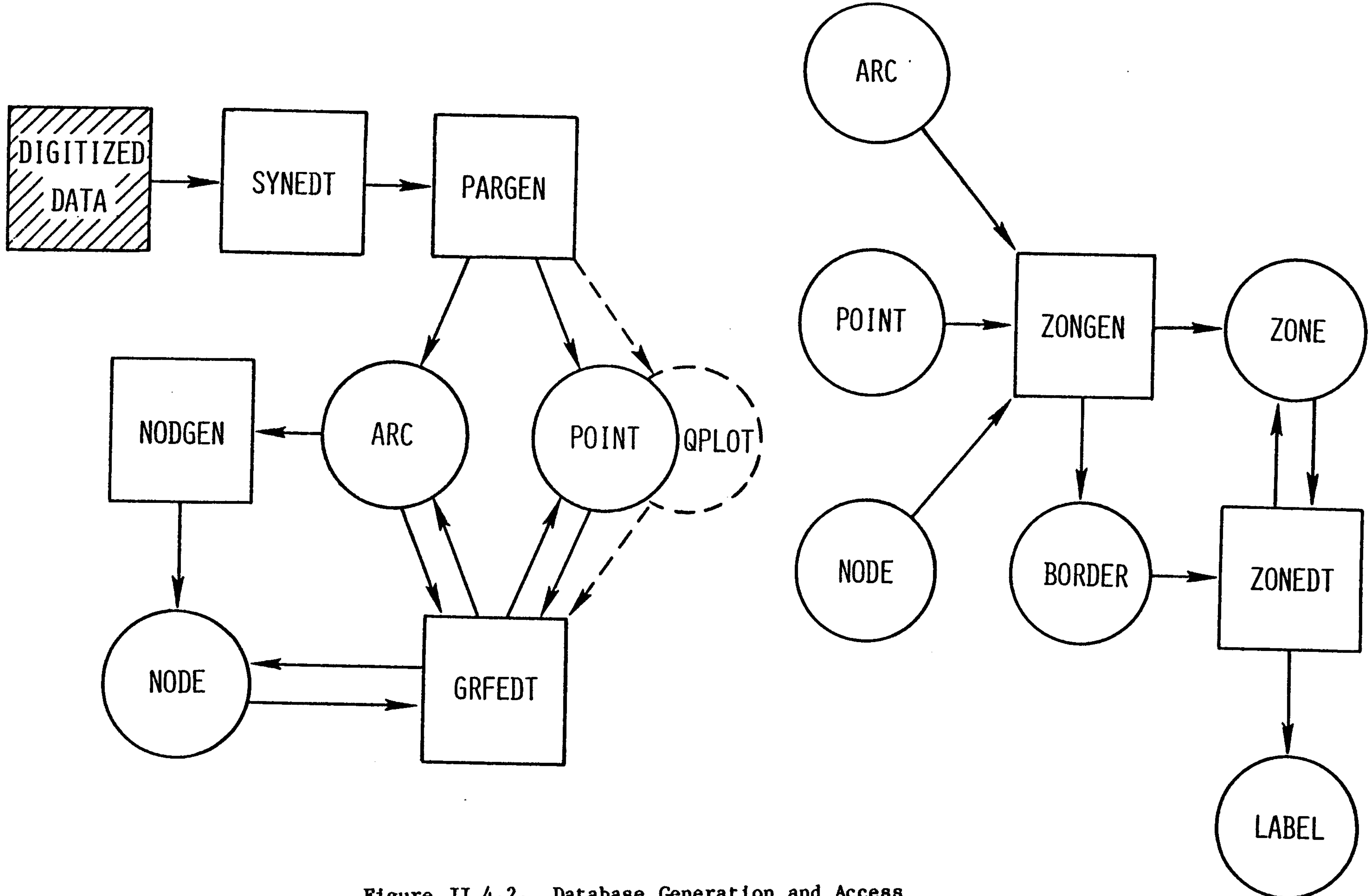


Figure II.4.2. Database Generation and Access database files are shown as circles. The arrows reflect the directional flow of data between programs and the database files. The structure and content of the base files is the subject of the next chapter.

II.5 Database Structures

The cartographic database employed by GIMMAP is a complex network consisting of six interconnected, random-access files called the Arc, Border, Label, Node, Point, Qplot, and Zone files. The relationships between these files are illustrated in Figure II.5.1. The connecting arrows indicate that each record in the originating file contains one or more pointers to records in the terminating file. The double arrows within a file indicate that there is some degree of intra-file record chaining.

Each of these base files consists of fixed-length records formatted to contain graphic and non-graphic information representing the map data. In some files, the records also contain pointers representing relationships between map features. The contents of the base file records are illustrated in Figures II.5.2 and II.5.3. The Qplot record format is not shown, but is parallel to the Point record format with x-y values corresponding to display coordinates for the graphics display terminal.

The Arc file is created by the PARGEN program, using the syntactically clean, digitized data file. Based on the rules of the syntax, PARGEN extracts the coordinates of the endpoints and the feature code for each arc. In Figure II.5.2., XFIRST and YFIRST represent the coordinates of the start node of the arc as it was digitized. XLAST, YLAST represent the coordinates of the end node. PARGEN places the interior points in the Point file and sets corresponding pointers in the arc record. The first pointer indicates the location of the x-coordinate of the first interior point following the start node. The second pointer indicates the location of the y-coordinate of the last interior point

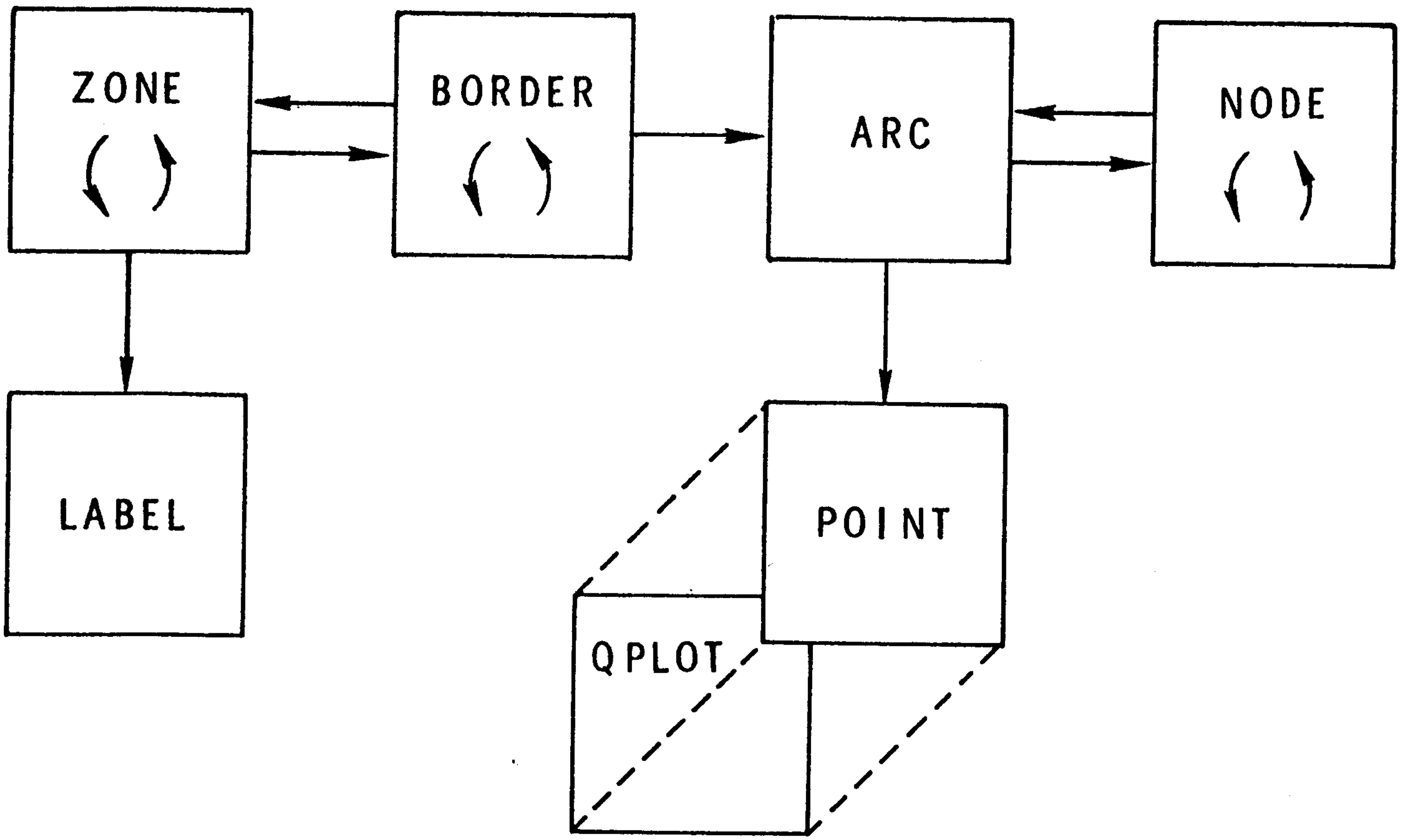


Figure II.5.1. Base File Relationships

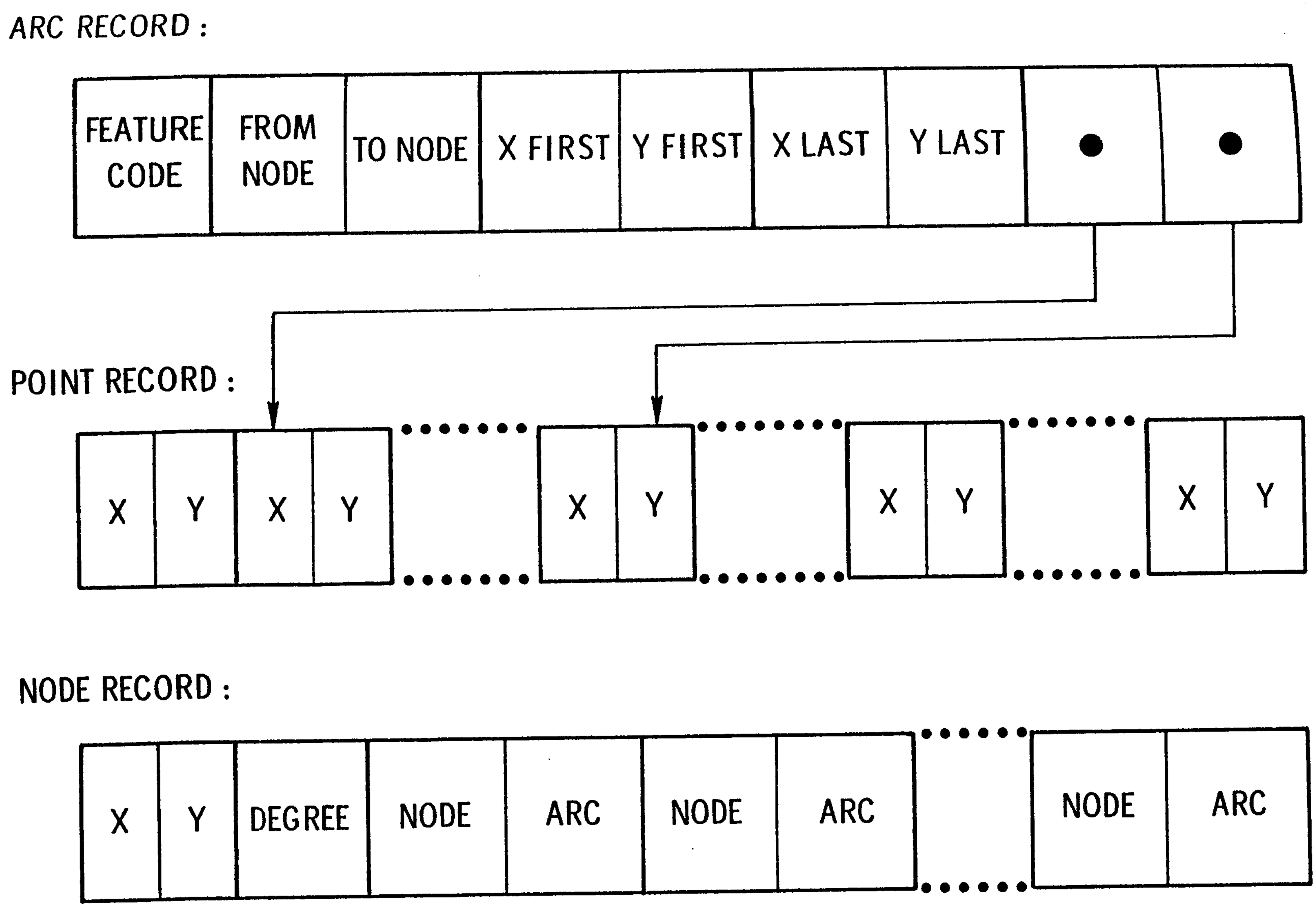


Figure II.5.2. Contents of Arc, Point, and Node Records

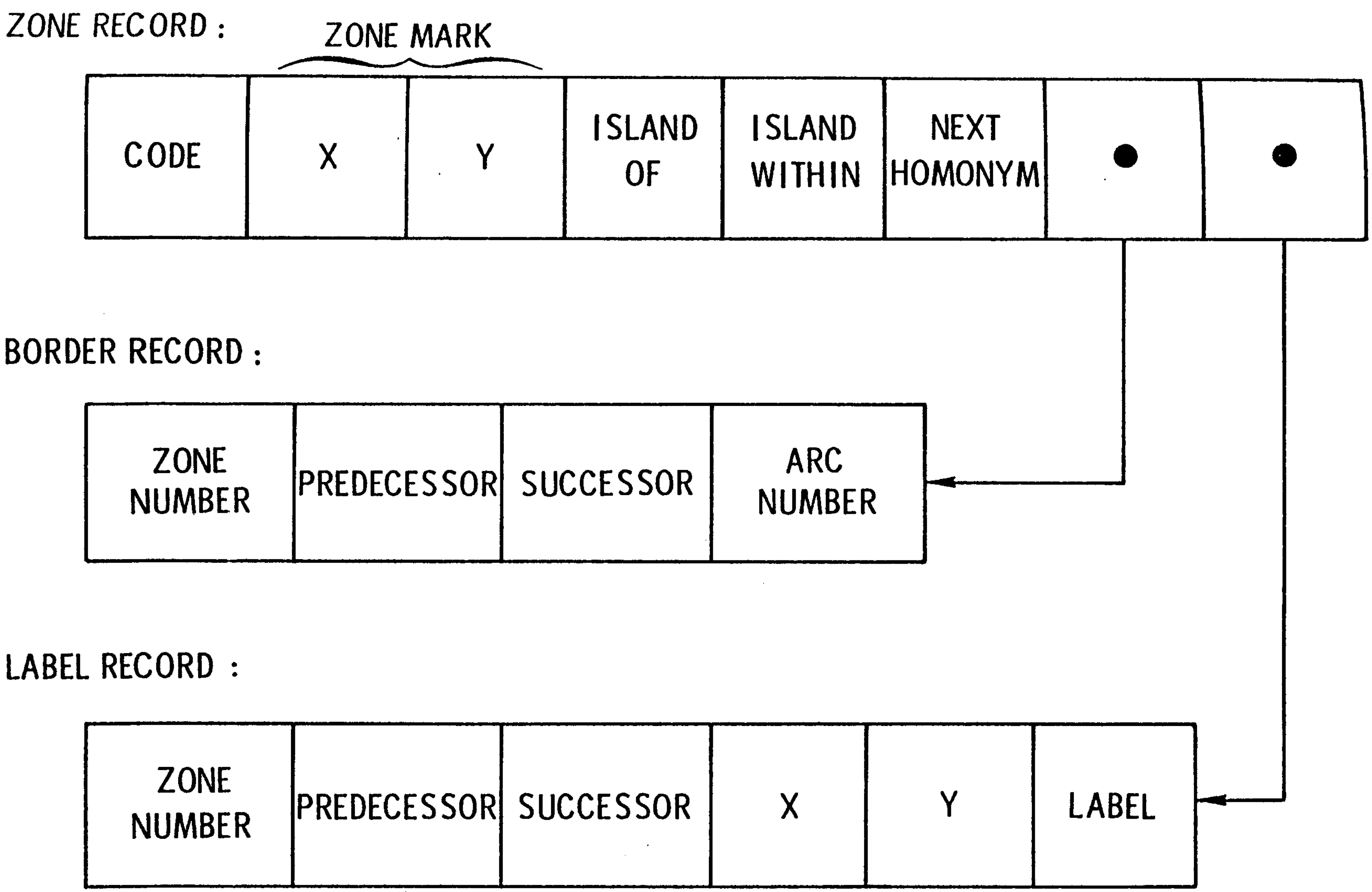


Figure II.5.3. Contents of Zone, Border, and Label Records preceding the end node. These coordinates are stored in the order in which they were digitized.

The node numbers corresponding to the endpoints of arcs are created by NODGEN following the execution of PARGEN. NODGEN creates the node records and then places the appropriate node numbers in all arc records. In each arc record, the FROM node is the node number of the start node, and the TO node is the number of the end node. These node numbers are the actual record numbers of the associated node records.

The interior point coordinates are packed into records in the Point file by PARGEN. Each point record contains eight pairs of x-y coordinates, beginning with an x coordinate. Pointers in the arc records indicate the locations of the first and last coordinates for the interior points of the arc, and are stored as absolute sequence numbers based on the order in which the coordinates enter the file. In the event that a file may contain too many points to be numbered (based on the size of the computer word), an additional number is contained in the arc record. This number is called the page number, and is zero unless the word size is exceeded by the number of coordinates in the file. In the event that an arc contains no interior points (such as for isolated points), the Point file pointers in the arc record are set to zero.

The Node file is created from the Arc file by the NODGEN program to describe the topological network contained in the map. Each node record represents a unique node corresponding to the intersection point of two or more arcs, or the endpoint of an arc which terminates within the map boundary without intersecting another arc at the termination point.

Each node record contains the x-y coordinates for its location, the number of intersecting arcs (DEGREE), and up to six node-arc pairs.

Each node-arc pair represents a connection between the two nodes by the arc named in the node-arc pair. The arc number of each node-arc pair is signed to indicate the start node and end node in the relationship. A positive arc number indicates that the node contained in the node-arc pair is the end node. Conversely, a negative arc number indicates that the node in the node-arc pair is the start node.

The node and arc numbers in the node-arc pairs of node records are the actual record numbers of the indicated nodes and arcs, provided that the arc number is interpreted as a positive value. The actual number of valid node-arc pairs in a node record is indicated by the degree of the node. All node-arc pairs beyond the degree of the node are set to zero by NODGEN.

The Zone and Border files are created from the Arc and Node files by the ZONGEN program using specifications by the operator to determine the subset of arcs to be considered for boundary generation. When initially created, the zone records contain only the pointers to the associated boundary arcs for the zone and a default zone or color code which is set to be equal to the zone number. Each boundary arc is described by a record in the Border file.

Each border record contains the associated zone number, pointers to the two border records which represent preceding and succeeding boundary arcs, and the arc number of the indicated boundary arc. For the first (last) boundary arc of a zone, the PREDECESSOR (SUCCESSOR) pointer points to the first (last) border arc itself. Since the zone boundary arcs are compiled in a counter-clockwise order, the arc number in the border records is signed to indicate that the arc is stored in the direction in which it is used (positive) or in the opposite sense (negative). Thus, each arc extracted for zone boundary generation is used in a positive sense for one zone and in a negative sense for the adjacent zone.

The ZONEDT program provides commands for the addition and updating of information in the Zone file. ZONEDT allows the operator to perform several functions interactively while viewing the zone information on the graphics display terminal. One such function is the selection of points to serve as locations for the placement of symbolic labels within zones. For each location and label entered, ZONEDT creates a record for the Label file containing the zone number, the symbolic information (LABEL) and the x-y coordinates of the selected point. All label records for a given zone are doubly-linked by PREDECESSOR and SUCCESSOR pointers.

The ZONEDT program allows the operator to update the Zone file with information vital to the selective retrieval process required for the production of edit plots and color separation materials for map making. The operator may select a point for each zone, called the zone mark, which is then used as the location to display the zone number of all zones. The zone or color code (CODE) may be changed to indicate the value of the major attribute represented by a zone. The presence of islands within surrounding zones is indicated by pointers (ISLAND OF and ISLAND WITHIN) set when the operator specifies these relationships. Finally, all zones of common color code may be linked (NEXT HOMONYM) for fast retrieval by color codes.

The random-access structure of the GIMMAP database allows immediate access to any record within any of the six base files. This structure provides the necessary support for interactive data entry, editing and update facilities provided in the GRFEDT and ZONEDT programs. In order to properly maintain the base files as records are added or discarded during these interactive processes, a Random File Management subsystem (RAFMAN) has been incorporated in GIMMAP. This subsystem is fully described in Section IV.

II.6 Interactive Graphical Editing

Computer-assisted graphical editing is the process of locating and correcting errors incurred in the transformation of information from a (graphical) source to the computer-usable representation, and verification of the results of the modifications. The GIMMAP interactive graphical editing facility, GRFEDT, provides the user with a convenient facility to locate and accurately correct errors in the graphical information stored in the database.

GRFEDT utilizes a graphics terminal to display any part (or all) of the information stored in the base files, with selection based on the preference for a specific area and/or a particular type of data. The correction of an error requires identification of the type of error and the exact item which is in error. The actual modification of data is initiated by the selection of a command from a menu-style list. After the command is acknowledged and specific identification of the item is completed, the potential modification is displayed and must be verified before an update is actually performed. The resulting data can then be displayed on command, preserving congruity between consecutive editing operations.

Types of Errors

The digitizing operator is responsible for converting a large set of data from visual form (map) into computer-usable form with the aid of the digitizing table and crosshair cursor. This tedious and time-consuming process inherently introduces error into the data; error usually credited to the human operator. Without a facility for the correction of these errors, the operator's efforts would be unusable.

The errors fall into two categories. First, there are errors in the preparation or execution of the guidelines established in the preparatory phase of the digitizing procedure. This category includes (1) duplicated, missing, or grossly incorrect arcs, (2) missing nodes, and (3) incorrect feature codes. The second category consists of manual accuracy errors including (1) incorrect interior point positioning, (2) non-matching nodes, (3) incorrect node positioning, and (4) unrecognized nodes.

Preparatory Errors

(1) Duplicated, Missing, or Incorrect Arcs. In the course of a digitizing session, the operator who does not proceed systematically may digitize a feature more than once, or may fail to digitize a feature altogether. Through carelessness, a feature may be digitized so poorly that it does not truly represent the original data.

(2) Missing Nodes and Bridging. Two or more arcs which legitimately share the same endpoints require the addition of an artificial node on one of the arcs. This "bridging" problem and other missing arc intersections may result from oversights by a poorly prepared operator.

(3) Incorrect Feature Codes. The failure to insert a header flag and accompanying feature code information at the proper junction in the digitizing process will lead to the attachment of improper feature codes within the database.

Accuracy Errors

(1) Incorrect Interior Point Positioning. Following a line with the cursor is, at best, a difficult operation. This fact, coupled with occasional hardware errors, produces some data points which differ significantly from the original data.

(2) Non-Matching Nodes. Digitizing a particular intersection point (node) on two or more separate occasions during a digitizing session will sometimes result in the creation of two nodes where one was intended, since the tolerance value used to automatically match such nodes must be minimized for cases in which two distinct nodes lie close together.

(3) Incorrect Node Positioning. As with interior points, nodes may be incorrectly positioned. (The distinction between node and interior point positioning lies in the procedures required for their correction.)

(4) Unrecognized Nodes. As described earlier (see Chapter II.3 on Data Input), the operator must not move the cursor when digitizing a point twice to represent a node, since the tolerance value for endpoint detection must remain small for digitizing sharply curving features such as rivers. If the two points digitized to represent a node are separated by a distance exceeding the tolerance value, the node will not be recognized by PARGEN.

Error Detection and Location

With few exceptions, detection of errors in the database relies heavily on visual examination by the operator. Duplicated arcs and uncorrected bridging problems are reported by NODGEN. Unmatched or unrecognized nodes and incorrect interior point positioning (of first interior points needed in the zone generation algorithm) may be reported subsequently by ZONGEN in the form of wrong turns or dead ends. Visual inspection is, however, greatly enhanced by the utility commands in the editing program.

The editing process begins with the location and display of a detected error. This process is facilitated by the use of a graphics display terminal on which any selected set of features from the data base may be accurately displayed. The selection process allows for the definition of an area to be displayed (windowing), and the selection of subsets of features by examination of feature codes (code sensitivity). All code-qualified features within the defined area are displayed in a fixed portion of the screen (the screen window) without distortion of graphical quality or relationship.

Windowing allows the operator to magnify an area of interest for close visual examination, while code sensitivity masks the display of unrelated features, thus clearing the image of unwanted obstructions. The automatic numbering of nodes assists in easy identification of specific features. These essential aids are discussed in a subsequent section on utility commands.

Error detection is enhanced by use of the "ten thousand plus" code which provides a dashed line display for re-digitized or appended features rather than the solid line display for other features.

Command Selection and Menu

Command selection level is the point at which the user selects one of 16 two-letter command mnemonics, initiating the execution of a specific program function. At program initialization and at the completion of any of the commands, the user is prompted with the terminal bell and a request to enter a command. The LM command lists a menu from which the user may select any of the following.

*** MENU FOR GRFEDT ***

BS	-	BELL SWITCH
CC	-	CHANGE CODE
CN	-	CHANGE NODE POSITION
CP	-	CHANGE (INTERIOR) POINT
DA	-	DELETE ARC
DW	-	DEFINE NEW WINDOW
LC	-	LIST (NODE) CONNECTIONS
LM	-	LIST THE MENU
MN	-	MATCH TWO NODES
NN	-	NUMBER NODES
PM	-	PLOT ENTIRE MAP
RP	-	REPLOT (same window)
SA	-	SPLIT ARC
SC	-	SELECT BACKGROUND CODE
ST	-	STOP EXECUTION
ZM	-	ZOOM WINDOW

The subsequent discussion on error correction refers to the specific editing commands listed above. Detailed descriptions of each of these functions are contained in Chapter III.5.

Error Correction and Verification

The editing program provides "modifying" commands which add, change, or delete information in the database. The selection of a particular command is dependent upon the type of error encountered. Specific identification of the feature is requested, usually in the form of the two node numbers identifying an arc. These nodes are then temporarily marked with a flashing hourglass symbol for verification of the selection. Further verification is requested by displaying the proposed modification for visual inspection by the operator, prior to updating the data.

Preparatory Error Correction

(1) Duplicated, Missing, or Incorrect Arcs. Duplicated or incorrectly digitized arcs must be eliminated from the database via the DA command. Missing arcs and re-digitized arcs may be added by use of the reentrance option of the PARGEN program. The use of the "ten-thousand-plus" code is recommended for these additions so that these features will be displayed by dashed (rather than solid) lines.

(2) Missing Nodes. Missing nodes may be created by splitting an arc into two arcs using the SA command.

(3) Incorrect Feature Codes. Feature codes may be changed by use of the CC command. When editing is complete for a "ten-thousand-plus" coded arc, the normal code may be regained with the CC command.

Accuracy Error Correction

(1) Incorrect Interior Point Positioning. The coordinates of any interior point may be changed to any other coordinates within the screen window by use of the graphics cursor within the CP command.

(2) Non-Matching Codes. Two nodes may be joined to form a single node using the MN command. The position of this node will be that of the first of the two original nodes. If this position is not correct, it may be modified as in (3) below.

(3) Incorrect Node Positioning. As with interior points the coordinates of nodes may be changed to any others within the screen window using the CN command.

(4) Unrecognized Nodes. From a correction standpoint, unrecognized nodes are equivalent to missing nodes and may be created using the SA command.

General Information and Definitions

The following information is presented as an introduction to terms and features of graphical editing which have not been previously defined. The detailed description of the GRFEDT program in Section III will refer to these terms to describe the operation of the GRFEDT program.

1. Screen and Virtual Windows

The screen window is the fixed portion of the graphics terminal screen on which the data are displayed. Its size, shape, and position are set by the program and do not change.

In contrast, the virtual or user window consists of a variable set of minimum and maximum x and y values (in user coordinates) for the two axes on which the data are based. These minima and maxima define a rectangle through which the data are filtered. Any line which is to be displayed is "clipped" at these boundaries. This user window is mapped onto the screen window and may be changed by the DW or ZM commands within the program.

2. Command Selection and Hardcopying

All input to the edit program is prompted by the output of explanatory information followed by the ringing of the control bell at the terminal (only if the bell switch is set on with the BS command). See the "Command Selection and Menu" section in this chapter for the list of all available commands.

At the discretion of the operator, the terminal screen may be hardcopied, provided that a hardcopy unit is available. This action should only be performed when the system is waiting for input (following the sound of the bell if the bell switch is on).

3. Graphical Input

The graphics terminal is equipped with a crosshair cursor (or other graphical input mechanism) through which the operator may communicate information to the program. The cursor consists of a vertical and a horizontal line on the screen. The positions of these lines determine a unique point on the screen and may be changed manually by thumbwheel controls (or by a joystick, if available) on the terminal console.

The location of the point on the screen is translated to user's coordinates (as defined by the program) and is used for identification or location of points and nodes. The operator must position the cursor properly and then type any character followed by a carriage return. The entry of the character and the carriage return cause the coordinates of the point and the entered character to be sent to the program. The character Ø (zero) is usually interpreted by GRFEDT as a request to return to command selection.

4. Standard MSR Termination Options

Most of the modifying functions terminate by offering three options to the operator. The M (MORE) option signals a desire to repeat the same command for a similar error, but without replotting the data to display the previous modification. The S (STOP) option requests a return to the command selection level without replotting the data. The R (REPLOT) option requests a return to the command selection level following a replot of the data to display the previous modifications. The existing window parameters and background code are not altered.

5. Code Sensitivity and the Background Code

All features in the database are associated with a feature code identifying the general type of the feature. This code is a four-digit integer and is used to mask features for editing as well as to selectively retieve information for display, plotting, and other functions of the GIMMAP system.

There is a special five-digit code available to enhance the identification of errors. This "ten-thousand-plus" code is simply the normal four-digit code which applies to the feature plus ten thousand. This code sensitivity extension provides a unique, dashed-line display for features which have been re-digitized or are being appended to the database.

Code sensitivity is implemented by use of a background code. All feature codes are masked against the background code in the following manner. Beginning with the rightmost (least-significant) digit, the feature code must match each non-zero digit of the background code. A zero digit in the background code will match any digit in the feature code. Thus a background code of zero will match all feature codes. If a feature code does not match the background code in all of the non-zero digits, the associated feature is rejected or masked.

This type of code sensitivity is used to determine which features will be displayed on the graphics screen as well as which features may or may not be modified by modifying commands. The proper selection of the background code allows the operator to edit a specific subset of the data, based on its feature type, and prevents improper modification of other types of data.

6. Conversation Area and Replotting

The screen window is designed to leave an area on the left side of the screen unused. This area is called the conversation area and is used for program/operator communications. All data entry and command prompting is printed in the conversation area. In the continual transitions from conversation mode to graphical mode, the progam maintains the position of the last conversational item printed. This facility recognizes a full conversation area; that is, when all the lines of the conversation area have been used, the program erases the screen and replots all information exactly as it had been displayed. Modifications which occurred since the previous plotting are reflected in the replotted display. Care should be exercised when nearing the end of the conversation area, so that useful information is not erased before it can be used.

7. Security

The security of the information contained in the database is maintained in a number of ways with respect to the editing program. These methods are outlined below.

(a) The editing program operates on a copy of the database known as the "working" files. These files may be copied to the "save" files (or vice-versa) using simple utility programs within GIMMAP. In the event that incorrect data are stored on the working files during the run of GRFEDT, these files may be reconstructed from the save files.

(b) Code sensitivity of the editing program prevents accidental modification of data unrelated to the error being corrected, and masks the display of extraneous features. The ten-thousand-plus coding for re-digitized data further extends this advantage.

(c) All modifying functions rely on record number verification and feature (point, arc, node) verification before performing any actual modifications to the database.

(d) Hardcopying of the terminal screen allows the operator to retain records of the modifications which have been performed. This precaution may circumvent costly regeneration of the data files by facilitating prompt discovery of operator-induced errors.

(e) The non-distortion algorithm used in the windowing process prevents confusion which might otherwise occur if the features from the database were graphically misrepresented in the display.

(f) The ability to magnify the image and to number the nodes in the display increases the operator's efficiency and effectiveness in properly correcting database errors.

(g) Extensive pre-modification checking helps to prevent the creation of database anomalies.

II.7 Hardware Considerations

The minimum hardware configuration necessary to support the GIMMAP system includes a minicomputer with 32K 16-bit words of memory; a disk storage device and interface with random-access control; a graphics display terminal and hardcopy unit with associated interfaces; a digitizing table with cursor, keyboard and interfaces; a plotting device with control unit and interface; and optional peripherals including a tape drive, system console, and line printer.

GIMMAP was initially developed at the Kansas Geological Survey with a hardware configuration including a Data General NOVA 1220 processor with 32K words, a floating-point processor and hardware multiply and divide functions; the Computer Equipment Corporation COMP-U-GRID digitizing system operating on a Bendix digitizing table; an Ampex DM323 disk storage device with 58 MB capacity; a Tektronix 4014-1 graphics terminal with Tektronix PLOT-1Ø software; a Varian 4211 printer/plotter and hardcopy unit; a Xynetics 1100 flat-bed plotter using a Hewlett-Packard C/64 controlling unit; and two Wang 1045, 9-track, 800 BPI tape drives.

The system has recently been upgraded by the replacement of the NOVA with a multiprogrammed Data General Eclipse S/230 computer with floating-point processor, memory mapping, and a total of 352 KB of memory (still at 32K per program). In adition, the new configuration contains a Data General 6061 disk with 192 MB capacity; a Data Printer Co. CT-1210, 750 CPM line printer; and several hardcopy and CRT terminals. The NOVA 1220 is now dedicated to controlling the digitizing operations.

SECTION III

CATALOG OF PROGRAMS

III.1 Introduction

This section provides descriptions of the program modules of the GIMMAP System. Programs are presented in alphabetic order, one chapter each. Relationships between programs are illustrated in the system flowchart in Chapter II.4.

Each chapter will contain the following:

A. Purpose

B. Definitions

C. Operation

D. Reports

E. Profile

    1. Name
    2. Subprograms
    3. Library Access
    4. Limits
    5. Base File Access
    6. Other Files

Keys to file code annotations:

= Input and output file

< Input file only

> Output file only

III.2 DIGGEN - Digitized Data File Generation

Purpose

This program produces a syntactically clean digitized data file containing data selected by the user from that available in the base files. This is a sequential file of x-y coordinate pairs, which adheres to the rules of syntax as outlined in Chapter II.3. This file will contain the results of any graphical editing, and hence could be used for tape archival, provided the base files are completely edited. Such a file may also be used to transfer data between installations or to regenerate base files following a catastrophic collapse.

Definitions

Each code string consists of a character (A or Z), followed by a comma, followed by a code range. The character is an "A" to specify arcs, or a "Z" to specify zones. The code range may be Ø to select all arcs or all zones. The code range may be a single negative number to specify a single arc or zone by its number. Finally, the code range may be a low and a high code separated by a hyphen to indicate a range of feature codes for arcs, or a range of zone colors (or codes) for zones.

Some examples of code strings follow:

A, 2200-2499	means all arcs with codes > 2200 but < 2499
Z, -25	means the single zone numbered 25
A, Ø	means all arcs

Operation

The operator is first asked to indicate if zones are to be extracted. This information allows the program to determine if the zone file is to be opened or not. Then, a set of instructions for arc/zone selection is given to the operator, who selects features based on code strings.

Code strings, as defined above, may be concatenated with commas as separators, and up to ten code strings may be entered for a particular run. An empty code string (carriage return only) indicates that all code strings have been entered. At this point, DIGGEN determines an offset transformation value for both the x and y coordinates, and a node matching threshold to determine when two interior points should be treated as a single node according to the syntax.

The operator is then allowed to enter a title or header for the output file, to identify the data for later processing. Once this is complete, the program selects features by making a pass through the zone or arc file for each applicable code string. A single header is thus produced for each set of arcs which matches the code string.

The one variation of the output file from a standard digitized data file is the existence of the zone color (code) as the y value of the arc header for all arcs retrieved for zones. The x coordinate of an arc header contains the feature code of the associated arcs, and for zones, the y coordinate will contain the color associated with the zone from which the arc is retrieved.

Reports

The program outputs the code ranges as they are being processed.

Program Profile

NAME: DIGGEN - Digitized Data File Generation

SUBPROGRAMS: CHKCD, DECIPH, FILOPN, GETARC, GETZON, HEAD, KODIN, OUTPUT, PROCA, PROCZ, SORT, TOKEN

LIBRARY ACCESS: RAFMAN

LIMITS: A maximum of 10 code strings

BASE FILE ACCESS: ARC, BORDER, POINT, ZONE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

Digitized Data File > 13

III.3 DIGPLT - Digitized Data Plotting

Purpose

The DIGPLT program is used to create a preliminary plot file from a syntactically clean, digitized data file (i.e., the output from SYNEDT). This data may then be plotted to check for gross errors caused by incorrect use of the digitizing control software, or by occasional failure of the digitizing hardware. The plot may also be used as a guide for future digitizing or initial preparation for graphical editing.

Definitions

The reader is referred to the GRFMAN and Data Input chapters for explanation of the terminology and procedures associated with the digitizing syntax and plotting.

Operation

DIGPLT begins by allowing the operator to select the input and output file names. The map header is read from the input file and displayed for operator verification. If it is incorrect, the operator may select another file or cancel the program execution.

DIGPLT then reads the control points from the input file and uses these values to determine the maximum dimensions of the plot. In this determination, an (effective) allowance of one inch is made for points which may exceed the limits imposed by the control points. All subsequent points are translated by one inch in both the x and y dimensions. Points which lie outside the window, as defined by the minimum and maximum plot dimensions, are ignored.

Headers provide the feature codes which are used to determine the plotting mode. Feature codes which are less than 2000 indicate isolated points, and feature codes greater than 2000 indicate arcs. Isolated points are plotted with the symbol "X" centered at the designated location. Arcs are plotted as sequences of connected straight line segments.

As indicated by the syntax (Chapter II.3), the flags result in the insertion of a "pen-up" command in the plot file.

At the end-of-file, DIGPLT inserts the appropriate control values to plot registration marks and terminate the plot. The number of points in the file is reported to the operator.

Reports

The number of points processed in the run is reported at the termination of the program. This number is useful as an approximation of plotting time and estimation of file initialization parameters.

Program Profile

NAME:	DIGPLT - Digitized Data File Plotting
SUBPROGRAMS:	FILOPN
LIBRARY ACCESS:	GRFMAN
LIMITS:	None
BASE FILE ACCESS:	None
OTHER FILES:	Terminal Input < 11
	Terminal Output > 10
	Digitized Data < 12
	Plot Output > 13

III. 4 GRFCHK - Graphical Checking

Purpose

Exceptional error conditions may cause inconsistencies in the base files representing the graph of the digitized information. If allowed to remain, these errors may propagate during further editing procedures. GRFCHK compares the node records to the arc records and reports any inconsistencies in connections or coordinate positions of nodes and arcs. This program is executed following the final construction of the base files by NODGEN, or whenever an exceptional condition is noted.

GRFCHK is a utility program which is not required in normal GIMMAP processing, but which may be useful for the development of special application extensions to GIMMAP. When inconsistencies are reported, the RAFPRN program may be used to determine the values of the records involved, and procedures may be developed to correct local errors.

Definitions

The reader is referred to the basic feature definitions in Chapter II.2 (Terminology) and the file structures in Chapter II.5 (Database Structures).

Operation

GRFCHK examines the Node and Arc files for the possible errors listed below:

Node Records

1. The degree of the node should equal the number of non-zero node-arc pairs in the record. These pairs should be stored in the first part of the record (left-justified) with no gaps.

2. Each node which is part of a node-arc pair should have a reciprocal node-arc pair in its record. These two pairs should be consistent in node number, and both should have the same arc number but with opposite signs.

3. Each arc which is part of a node-arc pair should have this node number in the proper position (From or To) in its record. This position is indicated by the sign of the arc in the node-arc pair.

4. If a node appears in two or more node-arc pairs of a single node record, then a bridging problem exists.

5. The coordinates of all nodes should lie within the rectangle defined by the minima and maxima values stored in the information record of the Point file.

Arc Records

1. The x and y coordinates for the first (From) node should equal the coordinates in the corresponding node record.

2. The x and y coordinates for the second (To) node should equal the coordinates in the corresponding node record.

3. The first (From) and second (To) node records should each contain a node-arc pair in which this arc appears, and the node of each pair should be the opposite node.

4. The endpoints of all arcs should fall within the minimum/-maximum criteria.

5. Island arcs which are not isolated points and which have no interior points are generally errors introduced by the syntax editor, and will be deleted.

GRFCHK performs this checking in batch mode and reports any discrepancies. The only correction that will be made by GRFCHK is setting arc endpoint coordinates equal to the coordinates in the corresponding record, if they are different. All such changes are reported.

Reports

GRFCHK makes specific reports for all errors and corrections which are described above.

Program Profile

NAME: GRFCHK - Graphical Checking

SUBPROGRAMS: CHKLIM, FILOPN, NAERR, NODARC, XYCHK, XYERR

LIBRARY ACCESS: None

LIMITS: The base files may contain a maximum of 3000 arcs and 3000 nodes.

BASE FILE ACCESS: ARC, NODE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

Optional Disk Output > 12

III.5 GRFEDT - Graphical Editing

Special Note: GRFEDT is the largest and most complex of the GIMMAP programs, hence the documentation for GRFEDT is considerably larger, and has been organized in a slightly different format than other program chapters.

Purpose

The interactive graphical editing facility, GRFEDT, provides the user with a convenient means to locate and accurately correct errors in the graphical information stored in the data base.

Computer-assisted graphical editing is a process of location and correction of errors incurred in the transformation of information from a (graphical) source to the computer-usable representation, and verification of the results of modifications.

GRFEDT utilizes a graphics terminal to display any part (or all) of the information stored in the base files, with selection based on the specification of a specific area and a particular type of data. The correction of an error requires identification of the type of error and the exact item which is in error.

The actual modification of data is initiated by the selection of a command from a menu-style list. After the command is acknowledged and specific identification of the item is completed, the potential modification is displayed and must be verified before an update is actually performed. The resulting data can then be displayed on command, preserving congruity between consecutive editing operations.

Command Selection and Menu

Command selection level is the point at which the user selects one of 16 two-letter command mnemonics, initializing the execution of a specific program function. At program initialization and at the completion of any of the commands, the user is prompted with the terminal bell (if on) and a request to enter a command. The LM command lists a summary (or menu) from which the user may select any of the following:

*** MENU FOR GRFEDT ***

BS	BELL SWITCH
CC	CHANGE CODE
CN	CHANGE NODE POSITION
CP	CHANGE (INTERIOR) POINT
DA	DELETE ARC
DW	DEFINE NEW WINDOW
LC	LIST (NODE) CONNECTION
LM	LIST THE MENU
MN	MATCH TWO NODES
NN	NUMBER NODES
PM	PLOT ENTIRE MAP
RP	REPLOT (same window)
SA	SPLIT ARC
SC	SELECT BACKGROUND CODE
ST	STOP EXECUTION
ZM	ZOOM WINDOW

The remainder of this chapter details the use of these commmands, which are divided into modifying and utility groups. The reader is urged to refer to Chapter II.6 for a general introduction to interactive graphical editing in GIMMAP.

Modifying Commands

The CC, CN, CP, DA, MN, and SA commands all cause modifications to the database upon successful completion. Documentation for each of these commands is presented in seven sections, as follows:

1. Error Description - A brief description of the type of error which is to be corrected.

2. Required Information - A list of information which must be supplied by the operator, usually to identify the error.

3. Procedure - An outline of the program-operator interaction involved in the correction.

4. Database Modification - A list of all updates performed upon completion of the command.

5. Termination Options - A description of procedural options available after successful completion of the command.

6. Possible Problems - A list of problems which may prevent successful completion of the command.

7. Comments - Specific operational suggestions.

NOTE: When "Standard MSR" appears under Termination Options, the reader is referred to "Standard Termination Options" in Chapter II.6.

CC - CHANGE FEATURE CODE

1. Error Description

The feature code of an arc is in error and the operator desires to change it.

2. Required Information

The node numbers identifying the arc.

3. Procedure

(a) Selection and verification of nodes identifying the arc.

(b) Replotting of arc for verification.

(c) Present code is reported.

(d) Selection of new code.

(e) Verification of change with arc number and new code.

4. Database Modifications

Feature code of affected arc record is updated.

5. Termination Options

Standard MSR.

6. Possible Problems

(a) Arc incorrectly identified.

(b) Background code prohibits selection of arc.

7. Comments

Background code should be properly set by the SC command. The NN command should be given to determine the identifying nodes of the arc. The conversion from a "ten-thousand-plus" code to the normal four-digit code (or vice-versa) is performed when the new code is selected to be -1, or 10000 plus the four-digit code.

CN - CHANGE NODE POSITION

1. Error Description

The location of a node is in error.

2. Required Information

The number of the node, as supplied by NN.

3. Procedure

(a) Selection and verification of the node.

(b) Cursor selection of the new position.

(c) Verification of the proposed new position.

4. Database Modifications

The coordinates of the affected node are set to the newly selected coordinates in the node file and in the arc records of all arcs sharing the node.

5. Termination Options

Standard MSR.

7. Comments

The operator may cancel the command during new position selection [step 3.(b)] by entering a 0 (zero) as the control character for the cursor.

CP - CHANGE INTERIOR POINT

1. Error Description

Due to operator error or table malfunction, an interior point position is incorrect.

2. Required Information

The node numbers identifying the arc which contains the point in error.

3. Procedure

(a) Arc selection and verification.

(b) Interior point selection and verification.

(c) Selection by cursor of new position.

(d) Verification of proposed position.

4. Database Modification

The coordinates of the point are changed in the Point file as well as their counterparts in the Qplot file.

5. Termination Options

At new position verification [step 3.(d)], the operator is offered three options similar to the standard MSR options. The difference is that Option 1 does not halt the command. Instead, it allows the operator to try again to select a new position. Options Ø (More) and 2 (Replot) are otherwise the same.

6. Possible Problems

An uncorrected bridging problem may cause the selection of the wrong arc. The operator must stop the command by entering zero as the cursor input control character and split one of the arcs using the SA command, or temporarily change the arc code of the other arc to a ten-thousand-plus code.

7. Comments

With the windowing capabilities, it is possible to achieve any level of accuracy desired when changing the position of a point. However, the operator must be aware that the windowing procedure will mask any arcs whose endpoints both lie outside the window boundaries. Further, it is recommended that the operator make an initial correction followed by a comparison plot of the data against the original document, before considering the correction to be accurate. The new position selection process may be terminated with the zero ("Ø") character used as the cursor input control character.

DA - DELETE ARC

1. Error Description

An arc has been digitized more than once, is no longer required in the data base, or has been re-digitized due to gross error.

2. Required Information

The node numbers identifying the arc.

3. Procedure

(a) Arc selection and verification.

(b) Replotting and verification of the arc.

4. Database Modifications

The arc record is returned to the freelist. The arc number is removed from the node records of its endpoints. If either node had degree of one, the node record is returned to the freelist.

5. Termination Options

Standard MSR.

6. Possible Problems

(a) Incorrect selection.

(b) Background code may prevent selection.

(c) As in CP, an incorrect bridging problem may cause the selection of the wrong arc.

7. Comments

This command is irreversible and should be used with caution.

MN - MATCH NODES

1. Error Description

Two nodes exist where there should be a single node. Usually detected visually following the NN command.

2. Required Information

The two-node numbers as supplied by NN.

3. Procedure

Node selection and verification (see Comments).

4. Database Modifications

The second node record is returned to the freelist after the arc records of all arcs connected to it have been updated to reflect connection to the first node (endpoint coordinates and node number), and after all node records of nodes connected by these arcs to the second have been updated to reflect connection to the first node.

5. Termination Options

Standard MSR.

6. Possible Problems

(a) The combined degrees of the two nodes may exceed the maximum limit for connected arcs (6). No matching may be performed without a deletion of an arc.

(b) The two nodes may be connected by an arc which must be deleted before matching can occur.

(c) The two nodes may both be connected to a third node which would create a bridging problem.

(d) Both nodes must have feature codes > 2000 (i.e., neither may be an isolated point).

7. Comments

Node selection requests both node numbers in a single entry. The first node number given must be that of the node which will remain intact. The second node will be matched to the first, hence the node whose position is most likely correct should be the first node given in the command.

SA - SPLIT ARC

1. <u>Error Description</u>

An interior point of an arc is to be changed to a node, thus splitting the arc into two.

2. <u>Required Information</u>

The node numbers identifying the arc and the interior point at which splitting is desired.

3. <u>Procedure</u>

(a) Selection and verification of the identifying nodes.

(b) Interior point selection and verification.

4. <u>Database Modifications</u>

The original arc record is changed to reflect the new endpoint node and (usually) a reduction in the set of interior points. A new arc record corresponding to the remainder of the points is created. A new node record is created for the new node located at the chosen interior point.

5. <u>Termination Options</u>

Standard MSR.

6. <u>Possible Problems</u>

(a) If an arc has no interior points, the actual location of the cursor (rather than that of the closest existing interior point) is used for the position of the new node.

(b) The feature code must be greater than 2000, that is, an isolated point may not be split.

7. <u>Comments</u>

This command may be required due to operator error or due to the addition of a new feature which connects to already existing arcs. Use of this command may be required also to resolve the bridging problem.

Utility Commands

The BS, DW, LC, LM, NN, PM, RP, SC, ST, and ZM commands provide utility functions to the operator, but do not modify the contents of the database. Documentation for these commands is organized in four sections, as follows:

1. General Description - A description of the utility and operation of the command.

2. Procedure - An outline of the program-operator interaction involved in the command.

3. Possible Problems - A list of special problems which may occur in the use of the command.

4. Comments - Specific operational suggestions and warnings concerning use of the command.

BS - BELL (ON/OFF) SWITCH

1. General Description

The graphic display terminal provides an audible bell which can be rung by the program. This bell is used to prompt user input or to command attention for a special display.

The BS command causes the bell to be turned off or on, depending on the previous state. The editing program begins with the switch off—the bell will not be sounded. If the user prefers the bell to sound, the BS command may be used.

2. Procedure

(a) Operator enters BS at the command selection level.

(b) Bell switch is flipped from off to on, or from on to off.

(c) Return to command selection level.

3. Possible Problems

None

4. Comments

The bell switch may be turned off or on at the operator's discretion.

DW - DEFINE WINDOW

1. General Description

This command is used to enlarge the image (while decreasing the scope) of the data displayed on the graphics terminal screen. Its use allows the operator to obtain a magnified view of a problem area. The portion of the screen used to display data is fixed in size and positioned so that when the operator selects a smaller, "sub-" portion with this command, the image of the data within the selected area is enlarged to fit the fixed display area. The operator must define a rectangle by using the cursor to select two opposite corners of the area desired. The order of selection is immaterial; the program will determine the order in which the corners have been selected.

2. Procedure

(a) Selection of two opposite corners of the chosen area by use of the cursor.

(b) Verification of window selection. Window selection is temporarily displayed on the screen. The operator accepts or rejects and tries again.

(c) Virtual window limits are reset to those selected.

(d) Data within the selected area is plotted on the fixed display portion of the screen, as filtered by the background code.

(e) Return to command selection level.

3. Possible Problems

(a) The operator may decide to retain the current window. This may be accomplished by entering Ø (zero) as the cursor control character.

(b) Successive windowing may be performed to enlarge a feature many times. However, unnecessary repetition of this command may cause run-time errors and result in an empty window or program abort.

(c) The operator may wish to select a window which does not entirely lie within the bounds of the current window. This may only be done with the ZM windowing command.

4. Comments

The algorithm used for re-plotting the data requires that one or both endpoints of an arc must reside within the chosen area for the arc to be plotted. Thus, careless use of this command may result in the temporary "disappearance" of arcs needed for identification and correction. The operator may use the NN and SA commands when necessary to prevent such disappearances. The area which is chosen must be such that the enlargement procedure does not distort the graphical representation of the data being displayed. In order to preserve this quality, the program will "grow" the size of the chosen area in either the x or y direction so that the y to x ratio is preserved. Hence, the operator will view no less than the area which was selected.

LM - LIST MENU

1. General Description

This command lists the command mnemonics and their definitions alphabetically.

2. Procedure

(a) The operator selects LM at command selection level and the command menu is displayed.

(b) Return to command selection level.

3. Possible Problems

None.

4. Comments

If the conversation area is nearly full, the display will be replotted before the menu is listed.

LC - LIST (NODE) CONNECTIONS

1. General Description

This command is used to determine the node numbers which identify the <u>opposite</u> endpoints of all arcs connected to a specified node. This becomes important when editing areas which are cluttered by many arcs, or when using certain commands (CP, SA), when only one node appears in the window.

The command cycles until a node number of zero is entered by the operator. This allows the connections of multiple nodes to be determined without typing the command mnemonic (MN) each time.

2. Procedure

(a) The operator selects a node by its number.

(b) If zero is selected, return to command selection.

(c) The node connections are reported and the user is prompted again--step (a).

3. Possible Problems

The operator must remember to enter a zero node number to exit this command before proceeding with the next command.

4. Comments

This command may also be used to discover invalid islands which are arcs with no interior points sharing a common node. These errors may occur as a result of syntax editing, and are reported by the GRFCHK program.

1. General Description

All nodes which reside within the current screen window will be numbered. Nodes which reside within the window but which are not connected to any arcs being displayed (some arcs may have been background code masked) will not be numbered.

This command is the single most important utility function used in recognition and identification of errors, since node numbers are assigned by the system and can only be determined by use of this command.

2. Procedure

(a) Nodes are automatically numbered.

(b) Return to command selection level.

3. Possible Problems

Numbers appearing at the edge of the screen window may be offset towards the center of the screen window.

4. Comments

(a) The use of this command is a prerequisite for most modifying commands, since most of those commands require one or two node numbers as identification of specific features.

(b) Execution of this command will result in immediate detection of the unmatched nodes. Errors of this type will generally be found where two (or more) numbers are clustered.

(c) The actual location of the node may be determined as the lower left-hand corner of the first digit of the node number (except as in 3. above). This fact is useful for the purpose of further windowing.

PM - PLOT ENTIRE MAP

1. General Description

This command is used to display all information in the data base with the exception of features masked by the background code. The window values will be reset to the minimum and maximum values corresponding to the entire data set for subsequent re-plotting.

Since this type of display requires no clipping of data at the screen window boundaries, a special routine and a display file are used to reduce the amount of time required to peform the plot. This procedure is therefore faster than that used for the windowing (DW and ZM) commands.

2. Procedure

(a) The screen is erased.

(b) The virtual window limit is reset to the maxima/minima values for the map.

(c) All data not masked by the background code are plotted on the fixed screen window.

(d) Return to command selection level.

3. Possible Problems

The operator may wish to reset the background code (using SC) prior to the use of this command.

4. Comments

Although the ZM command allows the operator to traverse the map without replotting, it is sometimes faster to use PM to locate an error following the use of DW or ZM commands and associated error correction. The data displayed by PM is complete, but due to the fixed size of the display screen, may be displayed at a scale which is inappropriate for accurate correction of errors.

RP - REPLOT (Same Window)

1. General Description

This command is used to replot the display data according to the current window limits and current background code. This command is used whenever the display has been somehow lost or damaged, or to prevent operator-programmer information loss during automatic replotting when the conversation area is full.

2. Procedure

(a) Current window limits are compared to the maxima/minima for the map.

(b) If within 5 percent, the virtual window limits are reset to the maxima/minima for the map.

(c) All features not masked by the background code are clipped if necessary and plotted.

(d) Return to command selection level.

3. Possible Problems

See "Possible Problems" under the DW, PM, and ZM commands.

4. <u>Comments</u>

Any updates occurring since the previous display was plotted will be shown in the new display.

SC - SELECT BACKGROUND CODE

1. General Description

This command is used to select the background code which masks the plotting of superfluous data as well as to prevent unwanted modification of features which are unrelated to an identified error.

2. Procedure

(a) The current background code is displayed.

(b) The operator selects a new background code of four (five for the 10000+ code) digits. A zero digit allows all codes to match at that digit. A zero background code matches all feature codes.

(c) Selection of the ten thousand (10000) code to display all ten-thousand-plus arcs is allowed by entering a negative value, such as -1.

3. Possible Problems

Incorrect code selection. The operator must understand the principle of using the background code.

4. <u>Comments</u>

Careful use of this command will assist the operator in correcting bridging problems with arcs which have differing feature codes.

ST - STOP EXECUTION

1. General Description

This command halts execution of the editing program and may be entered at the command selection level.

2. Procedure

Program halts.

3. Possible Problems

None

4. Comments

The graphics screen is not erased by this command and may be hardcopied following termination of the program.

ZM - ZOOM WINDOW

1. General Description

This command is an enhanced version of the DW command. It can be used to enlarge the image being displayed (by narrowing the scope of the display) or to enlarge the scope of the data being displayed (thereby shrinking the image). It allows the operator to select a single point as the center of the window being selected rather than two corner points as in DW. These enhancements allow the operator to "follow" a feature through the entire range of the data while requiring only a limited, relative specification of the window frame. Effective use of this command requires some experience.

2. Procedure

(a) The operator selects the window center with the cursor (using a command character of Ø to return to command selection).

(b) The selected center is displayed and verified.

(c) The operator enters a multiplication factor for calculation of the window size in relation to the current window. A factor of 1.Ø (or Ø) will retain the current window size. A factor less than 1.Ø (Ø.Ø1-Ø.99) will decrease the scope and enlarge the image. A factor greater than 1.Ø (1.1-99.9) will increase the scope and decrease the magnification accordingly.

(d) Verification of window selection.

3. Possible Problems

(a) This command allows the operator to move the viewing area outside of the current window, but within the scope of the data, without returning to the full view presented by PM. This advantage over the DW command requires a general knowledge of the entire data set in order to follow a selected feature without losing it from view.

(b) Selection of the window scaling factor requires practice, and is aided by the verification procedure.

4. Comments

Although this command is an improved version of the DW command, the DW command is retained. It is felt that the ability to clearly and completely define the entire window without requiring a scale factor is simpler, and is of value in the prevention of feature disappearance, as discussed in 4. under the DW command.

Verification of window selection requires the display of the new window, part or all of which may exceed the bounds of the current window. In this case, the current window boundaries are used to display the corresponding parts of the new window.

Program Profile

NAME: GRFEDT - Graphical Editing

SUBPROGRAMS: BNA, CC, CHKKOD, CHKNDS, CN, CP, DA, DW, FILOPN, FINDPT, LC, LM, MN, MSRQ, NN, NODARC, NULINE, PM, QPLOT, REPLOT, SA, SC, SELCOM, SHWNDO, SHWNEW, SHWPT, VPLOT, ZM

LIBRARY ACCESS: RAFMAN, GRFMAN

LIMITS: A maximum of 3000 arcs

BASE FILE ACCESS: ARC, NODE, POINT, QPLOT

OTHER FILES: Terminal Input < 11

Terminal Output > 10

III.5 NODGEN - Node File Generation

Purpose

The topological network of a map consists of the interconnections of the arcs at their endpoints. The connectivity information contained in this network is useful for zone creation, windowing and editing, grouping arcs, and other applications. Hence, a separate file called the Node File is created by the NODGEN program to comprehensively describe this network.

It is the responsibility of the digitizing operator to indicate the locations of nodes (arc intersections) by correctly digitizing the same coordinates (within a tolerance value) as the endpoints of all arcs which intersect at the node. NODGEN assembles the arc endpoints, and determines which coordinates represent the unique intersection point for each node. A record is created containing the node coordinates, the set of arcs which intersect at this node, and the set of nodes which are connected by these arcs to this node.

Definitions

The threshold or tolerance value is a distance selected by the operator, within which two distinct points are considered to represent the same point. In practice, this value is used to represent half the width of a square centered at the point in question. Any point lying within that square is considered to be the same, and the coordinates of the two points are averaged.

New arcs are those arcs which have been created by PARGEN, but whose endpoints have not been assigned to nodes by NODGEN. Initialization of the Node file causes all arcs to be treated as new arcs.

Duplicate arcs are those arcs which intersect each other at both endpoints; that is, they share the same pair of nodes at their endpoints.

An opposite node is a node which is the other endpoint of an arc connected to the node under consideration.

The degree of a node is the number of arcs which intersect at the node. The maximum degree is the maximum number of arcs which will be recognized at one node. This maximum node degree is six in GIMMAP.

Operation

NODGEN begins with operator selection of the threshold criteria for determining when two endpoints represent the same node. This value may vary depending on the accuracy of the digitizing equipment, the type of data, and the ability of the operator. This value should be large enough to allow NODGEN to recognize all arc endpoints which represent the same node, and it should be small enough to prevent distinct arc endpoints from being interpreted as the same node.

The operator is requested to select between initialization and non-initialization (re-entrance) of the Node file. If initialization is chosen, the data in the Node file is effectively erased, and the node numbers in the Arc file records are set to zero, thus causing all arcs to be treated as new arcs. If non-initialization is chosen, the files remain intact and only the new arc endpoints will be considered for node creation. If the threshold value is properly chosen and the digitizing operator is accurate, the resulting Node file will reflect the topological structure of the map.

The remainder of the run does not require the operator's presence, except, perhaps, to hardcopy the terminal output. NODGEN assembles the new arc endpoints and reconciles this set with any existing nodes. The coordinates of endpoints which match existing nodes are averaged with the existing coordinates. The resulting values are stored in the arc record and the node record. The node degree is increased by one and the corresponding opposite node and connecting arc numbers are added to the node record. All arc numbers are stored in the Node file as signed (+ or -) numbers where the sign indicates the starting node of the arc. The sign of an arc number reflects the chronological movement of the cursor by the digitizer operator. A negative indicates that the opposite node (the one in the node-arc pair) is the start node.

A new node record is created when an arc endpoint does not match any existing node. The degree is set to one, the opposite node and connecting arc numbers are entered in the node record, and the number of the new node is added to the arc record. The endpoint coordinates are used as the node coordinates, and the node record is added to the list of existing nodes.

As each new arc is processed, NODGEN reports the code, the arc number, and the two node numbers to the terminal. When all new arcs have been processed, a list of all new nodes and their node-arc connections is reported to the terminal followed by a report of run statistics.

Reports

Each new arc is reported with arc number, feature code, and node numbers, as the processing finishes for the arc. Each new node, or previous node whose connections have been changed, is reported with the node number and node-arc connections.

NODGEN will also report the occurence of duplicate arcs, and nodes with too many intersecting arcs (i.e., whose degree exceeds the maximum degree). The final report includes the count of new arcs and nodes, and a count of the total number of arcs and nodes.

Program Profile

NAME:	NODGEN - Node File Generation
SUBPROGRAMS:	FILOPN
LIBRARY ACCESS:	RAFMAN
LIMITS:	A maximum of 3000 nodes, or 2000 nodes if the projection option is used.
BASE FILE ACCESS:	ARC, NODE, POINT
OTHER FILES:	Terminal Input < 11 Terminal Output > 10

III.7 PARGEN - Point and Arc File Generation

Purpose

PARGEN generates the Point and Arc files from syntactically clean, digitized data using the rules of digitizing syntax. PARGEN may also be used also for the addition of new data to existing files. Secondary functions of PARGEN are to (1) store the control point coordinates in a reserved location of the Point file, (2) calculate the minimum and maximum coordinates and store this information in a reserved location of the Point file, (3) create a display file which parallels the information in the Point file, and (4) report statistics about the data in the Point and Arc files.

Definitions

No specific definitions will be given here. The reader may refer to the RAFMAN, GRFMAN, and Database Structures chapters for information concerning file initialization and maintenance, display file coordinates, and record contents for the Arc and Point files. The digitizing syntax terminology is discussed in the chapter on Data Input.

Operation

The operator is requested to select the name of the input file. PARGEN then reports the statistics of current usage of the data base (e.g., the number of arcs and points currently existing in the data base), and the operator may select to initialize the file or to append the new data through non-initialization. Initialization effectively erases all information in the Point and Arc files and should be used when first creating the database, or when the data existing in the database are deemed to be unusable or in some sense expendable. If initialization is selected, the operator must indicate the maximum number of records to be allowed in the Point and Arc files. In either case, parameters are set by PARGEN to properly maintain the file usage statistics.

In order for PARGEN to recognize arc endpoints (nodes), the operator must supply the threshold criteria to the program. This criteria represents the maximum distance (in both x and y) within which two points may be considered to be the same. The value is usually set to 1.5 times the accuracy of the digitizing hardware. For example, if the table accuracy is ∓ Ø.ØØ4, then the threshold criteria would be set at Ø.ØØ6. This allows for a slight error on the part of the operator or the table, but prevents the creation of numerous, false arc endpoints.

PARGEN performs the remainder of its work without operator intervention, except for possible hardcopying of the reported information. The rules of the syntax are used to determine the starting and ending nodes of each arc, the feature codes, and the proper sets of interior points. Each new arc is stored as a single record containing the feature code, the endpoint coordinates, and pointers to the interior points which are stored in the Point file. The feature code and endpoint coordinates of each arc are reported to the terminal.

When the input data are exhausted, PARGEN performs a number of tasks. Statistics from the current run and from the resultant database usage are stored in the database and reported to the terminal. The minimum and maximum coordinate values are stored with the control points in the second record of the Point file. A display file (the Qplot file) is created by converting the virtual (user) coordinates into display (screen) coordinates for the graphics display device.

Reports

The following items are included in the PARGEN termination report:

1. Number of New Arcs
2. Number of Total Arcs
3. Number of New Points
4. Number of Total Points
5. Number of New Point Records
6. Number of Total Point Records
7. Minimum/Maximum Coordinates
8. Next available space in the Point file.

Program Profile

NAME: PARGEN - Point and Arc File Generation

SUBPROGRAMS: ADDBUF, CREATQ, FILOPN, GETBAK, PAG, PUTARC, SETPTR, SETT, TRANS

LIBRARY ACCESS: RAFMAN

LIMITS: None

BASE FILE ACCESS: ARC, POINT, QPLOT

OTHER FILES: Terminal Input < 11
Terminal Output > 10
Digitized Data < 12

III. 8 PLTGEN - Plot File Generation

Purpose

PLTGEN provides the ability to produce a plot file from the information contained in the data base. The plot file may be used to produce simple plots for graphical editing of the data or to prepare scribecoats for color-separation material and eventual map printing. PLTGEN allows for selection by feature type and by feature code, and provides for combination of several selections within a given plot as well as several plots within a file. For example, the operator may select one plot to contain all rivers and all zones of code two. A second plot within the same file may contain all roads and all towns with a population less than a certain number.

Definitions

The following terms are defined here only to aid in the understanding of the operation of PLTGEN.

A feature type is the collection of all isolated points, all arcs, or all zones. An item is one member of a type—one specific point, one specific arc, or one specific zone. The item number is the corresponding arc number for points or arcs, and the zone number for zones. An item code is a feature (or arc) code for points and arcs, and a zone code for zones. A code range consists of a range of item codes (arc or zone codes), and is specified by:

LC - HC

where LC is the lower code number, and HC is the higher code number. The code range is used to select the group of item codes which (equal or) lie between the low and high values.

Operation

The operation of PLTGEN consists of an interactive and a non-interactive segment. The interactive segment allows the operator to select the data to be plotted, by methods described below. The information used for selection is stored in a file until all selection is completed. At that time, the non-interactive segment interprets the selection information, retrieves the appropriate data, and produces output to the plot file, which has been selected by the operator at the beginning of the run.

The interactive segment of PLTGEN follows the procedure listed below:

1. Selection of the (plot) output file.

2. Selection of x and y offsets for translation of multiple plots within a single file.

3. Selection of a scaling factor to be applied to both x and y coordinates.

4. Selection of feature type where

   Ø = All features

1 = Arc selection only

-1 = Zone selection only

5. Selection of individual items by number or code where the following selections are allowed

-N = Selection of a single item of the chosen type where N represents the item number.

Ø = Selection of all items of the chosen type.

C = Selection of the subset of items of the chosen type with item code = C.

LC-HC = Selection of the subset of items of the chosen type with item code within the range specified by LC-HC.

6. Option to return to code selection (5., above).

7. Option to add node numbering to the plot file. If selected, all nodes will be numbered.

8. Option to add map headers (titles) to the plot file. If selected, the operator may select the side (top, bottom, left, right) on which the title(s) will appear, as well as the information contained in the title(s).

9. Option to add a frame or border around the map area.

10. Option to add map registration marks.

11. The final selection is the option to repeat the above steps (2-10) for additional plots.

Reports

The non-interactive segment of PLTGEN produces a report which details the information (selection criteria) which was entered by the operator during the interactive selection segment.

Program Profile

NAME: PLTGEN - Plot File Generation

SUBPROGRAMS: ANGLE, BOX, BRKDWN, CHKLIM, DECO, FILOPN, FRAME, FRAMIT, HEAD, HEADER, INCH, INITA, LEFTRN, MARK, NUMBER, NZONKD, PLOTORDR, PLOTZ, PLTARC, PLTCLR, PLTFRM, PSETUP, RPLOT, TITLE, TOKEN, ZONKOD

LIBRARY ACCESS: GRFMAN

LIMITS: A maximum of 2000 total arcs or nodes selected for plotting.

BASE FILE ACCESS: ARC, BORDER, NODE, POINT, ZONE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

Plot Output > 12

Selection Information = 15

III.9 RAFPRN - Random File Printing

Purpose

RAFPRN is a utility program which provides easy access to the contents of the database files. The data in these files are stored in binary code requiring conversion to a character code (e.g., ASCII), and must be interpreted on the basis of the logical file structure if they are to be examined.

RAFPRN is useful to the programmer for the development of extensions to the GIMMAP software for special applications, and for recovery in the event of catastrophic database degradation due to system errors.

Definitions

The programmer should be familiar with the terminology concerning random files.

Operation

The complete set of data files (with the exception of the Qplot file) is available for inspection by RAFPRN. This includes the Arc, Border, Label, Node, Point, and Zone files. The operator selects a command from the menu indicating the file which is to be inspected. The respective commands are PA, PB, PL, PN, PP, and PZ.

The accounting information for the selected file is printed at the terminal, and the operator is asked to select the records in the file which are to be displayed. The selection consists of one or more record selections, separated by commas. A record selection is one of the following:

1. Ø = All records in the file.

2. N = Only record number N.

3. M-N = All records between (and including) M and N.

If an illegal record number is used, it will be interpreted as either the first or last record of the file, depending on which is appropriate.

The selected record contents are displayed and may be hardcopied if desired. At the completion of this listing, RAFPRN returns to the command selection level. The operator may issue further commands to inspect the same or other files, or may terminate the program with the ST (stop) command.

Reports

The accounting report for each file selected includes the freelist control values (see Chapter IV.1, RAFMAN) and any other pertinent information (which varies depending on the file). The contents of the selected records are reported with the corresponding record numbers. The operator must be familiar with the logical structure of the files (Chapter II.5, Database Structures) since labels are not included in the report.

Program Profile

NAME: RAFPRN - Random File Printing

SUBPROGRAMS: FILOPN, LIMIT, PRTARC, PRTBRD, PRTLAB, PRTNOD, PRTPTS, PRTZON

LIBRARY ACCESS: RAFMAN

LIMITS: None

BASE FILE ACCESS: ARC, BORDER, LABEL, NODE, POINT, ZONE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

Optional Disk Output > 4

III.10 SYNEDT - Syntax Editing

Purpose

A syntactically clean file of digitzed data (one without syntax errors) is required for input to the DIGPLT and PARGEN programs which, respectively, create a preliminary plot file, and create the Point and Arc files. The SYNEDT program is designed to locate and correct syntax errors in a file of raw digitized data.

SYNEDT applies the rules of the syntax (Chapter II.3, Data Input) to digitized data. When an error is found, SYNEDT (necessarily) makes certain assumptions about the cause of the error and corrects the error by insertion or deletion of data. The output file will be syntactically clean and can be used for data base construction or preliminary plotting.

Definitions

The terms which are relative to syntax editing are defined in Chapter II.3 on Data Input.

Operation

The SYNEDT program begins by offering the selection of input and output file names to the operator. The map header and control points from the selected input file are displayed, and the operator may reselect the input file, execute the program using the selected file, or abort the program.

The operator is given a suggested tolerance value (see Definitions in Chapter III.6) and may accept or reset this value. The remainder of the run is in batch mode. The operator's presence is not required, except to hardcopy the program reports if desired.

SYNEDT processes the data sequentially, always maintaining a buffer which allows for deletion of data which has been previously processed. Errors are corrected as they are encountered, and each new feature code and each new arc is reported to the terminal. At end-of-file, a run summary is reported to the terminal.

Reports

SYNEDT reports the feature code whenever a header is encountered in the data. As each new arc is discovered, SYNEDT reports the coordinates of the first point of the arc, and a count of the error corrections and point deletions that occurred since the previous arc began.

At the conclusion of the run, SYNEDT makes a report of information gathered during the run. The counts of data types may aid the operator in selecting the size of the files for data base creation. The error information may be used to evaluate and instruct the operator. The report contains counts of the following items:

1. Isolated Points
2. Arcs
3. Nodes
4. Interior Points
5. Error Corrections
6. Point Deletions
7. SYNEDT operations.

Program Profile

NAME: SYNEDT - Syntax Editing

SUBPROGRAMS: ACT, FILOPN, INIT, INP, OUTP, REPORT, SHIFT, TERM, TOKEN, TRANS

LIBRARY ACCESS; None

LIMITS: None

BASE FILE ACCESS: None

OTHER FILES: Terminal Input < 11

Terminal Output > 10

Digitized Data < 12

Cleaned Data > 13

III.11 ZONEDT - Zone Editing

Purpose

The ZONEDT program provides commands for interactive updating of zones and zone boundaries (borders). With these commands, the operator may perform the following editing functions:

1. Selection of zone indentification marks.
2. Selection of zone label marks.
3. Modification of zone color codes.
4. Inclusion of island zones in surrounding zones.
5. Linking zones of similar zone color codes.
6. Deletion of zones.

In addition, several utility commands provide selective retrieval and display of zones which are helpful in performing the functions listed above.

Definitions

Definitions of terms related to zones may be found in Chapter II.2. The use of the display terminal is described in Chapter II.6.

Modifying Commands

DZ = DELETE ZONE

The operator selects a zone by the zone number. The zone boundary is displayed and the operator must verify that the proper selection has been made. The zone is then deleted from the database and cannot be returned without the use of the ZONGEN program.

LH = LINK HOMONYMS

This command results in automatic linkage of all zones by zone codes, allowing for efficient retrieval (by these codes) for production of color-separation materials and other retrieval functions. No operator input is required. This command should be preceded by selection of zone codes with the MZ command.

LI = LINK ISLANDS

In order to provide complete retrieval of boundaries for zones which have islands, the operator must specify which island zones lie within corresponding "surrounding" zones, as follows: (1) The operator must display the zone boundaries using the PA command, (2) display, the zone numbers using the PM command, (3) select a "surrounding" zone by number, (4) supply the zone numbers of all islands included in the surrounding zone, (5) enter zero as the zone number to signal that all islands have been entered, (5) go to Step 3, if necessary, and (6) when all surrounding zones have been completed, enter a null zone number to terminate the command and return to the command selection level.

LZ = LABEL ZONES

The operator must display the zones and zone numbers using the PA and PM commands. The operator then selects a zone by its zone number, entering a zero to terminate the command. When a zone is selected, the operator may select a maximum of five locations for the placement of zone labels (using the graphics cursor), and the labels which are to be placed there. For very small zones, a hook may be placed which connects a point within the zone to the location of the label (outside the zone). The selection of label points is terminated by entering a zero cursor control character (see Chapter II.6).

MZ = MARK ZONES

The operator must first select which zones are to be marked. The selection is between all zones, or only those zones which have never been marked. The operator then selects whether or not to change the zone color codes during the selection of zone marks. These codes are initially set by the ZONGEN program and may be updated with the MZ command. When these parameters have been set, the zones are displayed one at a time, and the operator selects a single point within the zone using the graphics cursor. The command may be terminated by the entry of zero as the cursor control character. If the operator has chosen to update the zone color codes, the program will request the new code following selection of the zone mark. When all (selected) zones have been marked (or when the operator terminates the command with the zero cursor control character), the program returns to the command selection level.

Utility Commands

LM = LIST MENU

The list of commands is displayed at the terminal.

PA = PLOT ALL ZONES

The screen is erased and all zone boundaries are displayed. A background code is set so that subsequent execution of the PM command will display marks (or labels) for all zones.

PC = PLOT BY CODE

The operator selects a zone code. The boundaries of all zones which match this code are displayed, hence the LH command should be executed before attempting to plot by code. The background code is set so that subsequent execution of the PM command will display marks (or labels) for this selected set of zones only.

PM = PLOT MARKS

The operator selects either zone marks or label marks for display. If zone marks are selected, the operator must select either zone numbers or zone color codes to be displayed at the zone marks. Labels are displayed with the zone label marks. As indicated above, only those zones which were most recently displayed (by PA or PC) will be marked or labeled.

ST = STOP EXECUTION

The program stops without erasing the screen, so that the operator may hardcopy the screen if desired.

Program Profile

NAME: ZONEDT - Zone Editing

SUBPROGRAMS: BNA, DZ, FILOPN, GETZON, INKOD, LETTER, LH, LI, LM, LZ, MZ, NULINE, PA, PC, PLTALL, PM, QPLOTZ, SELCOM, SHWOLD

LIBRARY ACCESS: RAFMAN, GRFMAN

LIMITS: None

BASE FILE ACCESS: ARC, BORDER, LABEL, POINT, QPLOT, ZONE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

III.12 - ZONGEN - Zone and Border File Generation

Purpose

ZONGEN is responsible for assembling the boundaries of zones from the data in the Arc, Node, and Point files. The left-turn procedure begins at a node (the start node), follows one of the arcs incident at this node, and takes the left-most arc at each successive node until returning to the start node. This procedure is illustrated in Figure III.12.1.

Each series of arcs forming a zone boundary (closed curve) is stored in a linked list in the Border file. Additional information about the zone is stored in a record in the Zone file.

Definitions

The left-turn boundary generation procedure uses the first interior point of an arc to calculate an angle for comparison. If this point is erroneously placed, it may result in a wrong turn. This type of error and its correction have been illustrated in Chapter II.5. Wrong turns may also be the result of missing data. In this case, the missing arc must be digitized and edited before ZONGEN will produce the complete set of zones.

The number of zone boundary arcs which may be assembled for one zone is arbitrarily limited by a maximum (usually 200). If the zone generating procedure attempts to create a zone whose boundary exceeds the limit, a zone overflow message will be given, and the assembly of that zone is terminated.

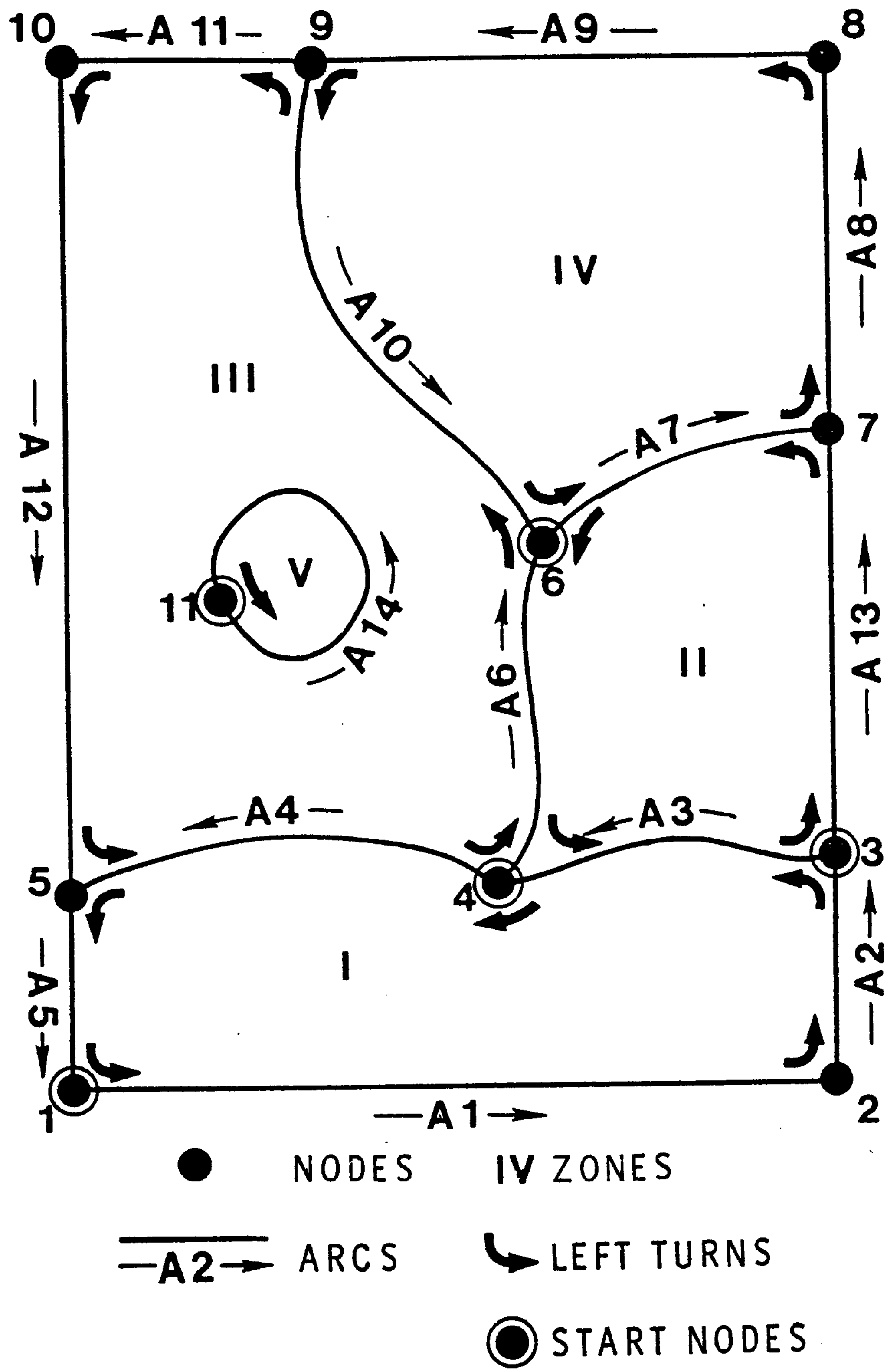


Figure III.12.1. Zone Boundary Generation

Certain errors in the data may cause the zone generating procedure to reach a node from which there is no possible exit arc. This situation is termed a dead end.

Operation

ZONGEN begins by allowing the operator to select one or two ranges of feature codes to be used for filtering arcs in the zone generation process. Each range consists of a lower and upper bound for feature codes, and only those arcs whose feature codes lie within one or both of the ranges will be considered for zone boundary assembly.

ZONGEN creates a modified image of the Node file containing only those node-arc pairs whose arcs meet the code selection requirements, and only those nodes which are endpoints of a least one such arc. All other nodes and node-arc pairs are ignored.

Once this image has been constructed, ZONGEN proceeds to apply the left-turn procedure to each (modified) node record. As node-arc pairs are used in zone boundaries, they are marked so that each will be used for only one zone. Each arc is contained in the image as both a positive and negative number (each arc is a boundary between two zones), thus each arc will be used twice; once in each direction.

When a zone boundary is completed, a zone code equaling the zone number is assigned. A record containing information about the zone (see Chapter II.4 on Database Structures) and linkage to the zone boundary arcs is created and stored in the Zone file. The (signed) arc numbers which comprise the zone boundary are linked and stored in the Border file. A positive arc number indicates that the arc is being used in the zone boundary in the same direction as it was digitized.

Reports

ZONGEN reports the zone number, all node and arc numbers used in the boundary, and the number of arcs in the boundary for each zone.

Missing data or other errors may cause wrong turns resulting in dead ends or zone overflow. This information is also reported to the terminal. If present, these errors must be corrected before ZONGEN may correctly assemble all zone boundaries.

At the end of the run, ZONGEN reports the total number of zones, the number of dead ends, and the number of zone overflows.

Program Profile

NAME: ZONGEN - Zone Generation

SUBPROGRAMS: ANGLE, CHANGE, DEND, FILOPN, FILTER, KODIN, LEFTRN, LOAD, MOSAIC, OVRFLW, SHOW, ZONOUT

LIBRARY ACCESS: RAFMAN

LIMITS: A maximum of 200 zones, 700 border arcs per zone, 750 filtered nodes and 3000 arcs.

BASE FILE ACCESS: ARC, BORDER, NODE, POINT, ZONE

OTHER FILES: Terminal Input < 11

Terminal Output > 10

SECTION IV

UTILITY LIBRARIES

IV.1 Introduction

The two utility libraries, RAFMAN for Random File Management, and GRFMAN for Graphics Management, provide the basic support required by GIMMAP to control and display data.

RAFMAN provides a uniform and consistent approach to the acquisition, access control, and deletion of data common to all GIMMAP base files. GRFMAN performs a parallel function for the display, graphical manipulation, and retrieval selection functions common to most graphical GIMMAP programs.

This section provides an overview of the logical structure of the random files used in GIMMAP, and catalogues the complete sets of routines for both RAFMAN and GRFMAN.

IV.2 RAFMAN - Random File Management

Introduction

The GIMMAP system requires that the basic data files be of such a nature that any record can be accessed directly, and hence, the system is designed to use random files. Without random files, a record could only be accessed by searching (on the average) a majority of the file for a record with a particular key. This sequential process is costly, both in time and resources, and is circumvented by using random files.

The GIMMAP system also requires the capability of adding or deleting any given record, but has no concern for the chronological order in which additions and deletions occur. These facts suggest that a file management program would be feasible and desirable. Hence, at the least, one word of each deleted (or unused) record should be used to distinguish good (used) records from free (unused or deleted) records. This flagging process is one major feature of RAFMAN, the Random File Manager.

The second major feature is the linking of all free records in a doubly-linked chain. This is accomplished by using two words of each free record as pointers to the preceding and succeeding records in this "freelist." This feature prevents the need for file compaction which could become necessary after many records within the file were deleted. It also eliminates the necessity of searching the file for freelist (available) records when a new record is to be added.

The third feature of RAFMAN is the accounting record. This record is the first record of the file and contains a general picture of the file for easy inspection. It includes five integer values:

FIRST - A pointer to the start of the freelist

LAST - A pointer to the end of the freelist

MAX - The maximum length of the file in records

USED - The number of good or used records in the file

LREC - The length of the record in words

Use of this bookkeeping record is generally reserved for RAFMAN, and the above values are rarely altered by other programs.

File Structure

Each GIMMAP random file consists of three parts. The first part (A) consists of a record which contains accounting information for RAFMAN as described below. The second part (B) is the set of records which are currently used by the GIMMAP system. These records are not necessarily contiguous and their quantity is contained in the accounting record. The third part (C) consists of all other records in the file. This part is called the "freelist" since all records within it are free to be used by the system. All records in the freelist are doubly-linked by chain pointers.

(A) Accounting Record

The first record of the file is reserved for the accounting information. This accounting record contains five integer values and should not contain any others. These values are:

1) FIRST - The record number of the first available record in the freelist chain.

2) LAST - The record number of the last available record in the freelist chain.

3) MAX - The maximum number of records in the file (as set by RMINIT), limited by file creation parameters.

4) USED - The number of records in the file which do not belong to the freelist, that is, are being used by the system.

5) LREC - The fixed length (in words) of all records in the file.

(B) Used Records

There are only two restrictions on records used in the file when structured to use RAFMAN.

1. The record length must be set to at least five words to contain the values in the accounting record.

2. The first word used in a record should be a positive integer value. This is true if it is necessary for the system to distinguish between freelist records and used records (see below).

(C) Freelist

The freelist is a doubly-linked chain of records which are not currently used by the system. The advantage to having such a chain is that used records may be released anywhere in the file and reclaimed without the need for compaction. A freelist record consists of three integer values:

1) FLAG - The first word of a freelist record will contain a -1 and may be used to distinguish freelist records from used records.

2) PRED - Contains the record number of the predecessor in the freelist chain.

3) SUCC - Contains the record number of the successor in the freelist chain.

The first and last records in the freelist are special since the first has no precedessor and the last has no successor. Hence, the first record in the chain has itself as a predecessor and the last record has itself as a successor.

RAFMAN Subroutines

There are seven routines which support the random file management in GIMMAP. These routines and appropriate calling sequences are described below.

RMCHKR - Check Record Number

This routine provides protection against access to records which lie outside the boundaries defined in the accounting record. Only those records which lie between the fixed accounting record and the last available (MAX) record may be legitimately accessed. In the case of an error, the RMERRO routine is called to report the appropriate information, and the status flag is set to inform the calling routine of the error. It should be noted that the calling routine is responsible for determining if the selected record is a part of the freelist.

CALL RMCHKR (IREC, MAX, IFC, ISTAT), where

IREC = Record number

MAX = Maximum allowed record number

IFC = File code of associated file

ISTAT = 1, if IREC is legitimate

-1, otherwise.

RMERRO - Error Report

This routine reports a variety of errors related to random file management to the output file previously indicated to the RMIOFC routine. A call to RMERRO is made whenever any illegal record access is attempted, or when some inconsistency is found in the file. RMERRO prints a message containing the file code of the affected file and a predefined error code indicating the type of error. If this error occurs during an operation on the graphic display terminal, a hardcopy command will be issued to the terminal.

CALL RMERRO (ICODE, IFC), where

ICODE = Error code to be reported

IFC = File code of associated file.

RMGETR - Get a Record

This routine is called to obtain a record from the freelist. Provided that a record exists (the file is not full), the first available record is unchained from the freelist. If there is no available record, an error message is printed by RMERRO. Both conditions are indicated to the calling program which is responsible for resetting the freelist flag in the record.

CALL RMGETR (IFC, NUM), where

IFC = File code of affected file

NUM = Record number of the first available record in the freelist

= -1 if record is not freelist flagged.

RMINIT - Initialize a Random File

This routine initializes the account record contents and the free-list structure for the indicated file. The operator is informed of the current file size and the suggested size from the calling program. The operator then selects a maximum size for the file, and the appropriate record values are reset.

This routine is also called to extend the size of an existing file without restructuring the freelist. This occurs when RMGETR attempts to obtain a record from a full file, and causes the file MAX to be increased by 25 percent of its current value.

CALL RMINIT (IFC, MAXP, LENGR), where

IFC = File code of the file to be initialized

MAXP = Suggested maximum number of records

LENGR = Length of the records in words.

NOTE: The operator will be allowed to select the maximum record number based on the contents of the file and the value suggested in the call.

RMIOFC - Set Output File Code

This routine is called to inform the RAFMAN subsystem of the output file code for terminal output in case of error. This routine must be the first RAFMAN routine to be called.

CALL RMIOFC (IFC), where

IFC = The output file code.

RMPAGR - Page a Record (Point File Only)

This routine decodes a page number and datum offset number into a record number and intra-record offset value. It is used by GIMMAP programs to determine the location of coordinates in the Point File, which is a packed, paged structure.

CALL RMPAGR (NPAG, NUM, IREC, IPTR, ISIZ) where

NPAG = Page number containing the record

NUM = Datum number of item to be paged

IREC = Record number containing item

IPTR = Offset within the record

ISIZ = Number of data items within a record.

RMPUTR - Put a Record in the Freelist

This routine returns a record to the freelist. The freelist is restructured so that the most recently returned record is set to be the FIRST freelist record available. If the freelist flag is already set, an error is reported, and the calling routine is informed.

CALL RMPUTR (IRC, NUM), where

IFC = File code of affected file

NUM = Record number of record to be returned to the freelist

= -1 if record is already freelist flagged.

IV.3 GRFMAN - Graphics Management

The graphics management routines, collectively called GRFMAN, provide graphical display capabilities for the GIMMAP system. These routines are used to support interactive editing and retrieval functions on the graphic display (CRT) terminal, and to interpret base file information for the production of intermediate edit plots and scribe-coat preparation for eventual four-color map production. These routines are logically divided into three groups.

The first group provides direct display capabilities on the graphic display terminal, and is called "unbuffered." This group is further divided into three categories—control, virtual, and screen, as indicated by the first three letters of the routine names. Control routines modify the state of the display device, (such as erasing the contents). Virtual routines perform various display functions, (such as drawing a line), and operate on user-defined "virtual" coordinates. Screen routines parallel the virtual routines, but operate on the fixed "screen" coordinates defined for the display terminal.

The second group of routines provides "buffered" display functions, where buffering refers to the fact that the collection of display-driving data is separate from the actual execution of the display itself. This buffering is used when very large sets of data are to be displayed, and virtually quadruples the speed of the display. This group is subdivided into control, virtual, and screen categories, similar in function to those of the unbuffered group.

The third group of routines is the plotting routines. This group provides the ability to convert data to a format capable of driving the plotter to produce paper copy on scribecoat output. There is no subdivision of this group, because the coordinates of data in the GIMMAP base files are equivalent to those expected by the plotter.

The remainder of this chapter provides a catalogue of the GRFMAN routines, a very brief explanation of the functions of each, and a description of the calling sequences and parameters for each routine.

GRFMAN Unbuffered Routines

Operation	Control	Virtual	Screen
Set mode to alpha-numeric	GMCAMD		
Ring terminal bell	GMCBEL		
Set character size	GMCCSZ		
Erase screen, set alpha mode	GMCERA		
Finish use of graphics routines	GMCFIN		
Set graph mode (or pen up)	GMCGMD		
Hardcopy terminal screen	GMCHCP		
Initialize graphics routines	GMCINI		
Advance alpha cursor (new line)	GMCNUL		
Set Z mode (write thru, etc.)	GMCZMD		
Draw a box		GMVBOX	GMSBOX
Clip a line segment		GMVCLP	GMSCLP
Calculate clip code		GMVCOD	GMSCOD
Cursor input		GMVCSR	GMSCSR
Convert virtual-screen		GMVCVT	GMSCVT
Draw solid line segment		GMVDRW	GMSDRW
Draw dashed line segment		GMVDSH	GMSDSH
Draw window frame		GMVFRM	GMSFRM
Draw hourglass figure at point		GMVHRG	GMSHRG
Return window limits		GMVLIM	GMSLIM
Draw dark line (move pen up)		GMVMOV	GMSMOV
Print number at point		GMVNUM	GMSNUM

Draw plus symbol at point	GMVPLS	GMSPLS
Set window limits	GMVWND	GMSWND
Draw "X" symbol at point	GMVXIT	GMSXIT

GRFMAN Buffered Routines

Operation	Control	Virtual	Screen
Set dashed mode in buffer	GMBDMD		
Set graph mode in buffer	GMBGMD		
Plot buffer contents	GMBPLT		
Set Z mode in buffer	GMBZMD		
Clip line segment		GMVCLP	GMSCLP
Calculate clip code		GMVCOD	GMSCOD
Add line segment (point) to buffer		GMBVLN	GMBSLN
Add point symbol to buffer		GMBVPT	GMBSPT

GRFMAN Plotter Routines

Draw a line to a point	GMPDRW
Finish plotting commands	GMPFIN
Initialize plotting	GMPINI
Move to a point	GMPMOV
Draw plus symbol at point	GMPPLS
Draw an X at a point	GMPXIT

Buffered Control Routines (GMB---)

GMBGMD - Set Graphics Mode

CALL GMBGMD

GMBPLT - Empty the Plot Buffer*

CALL GMBPLT

*The actual emptying process may be performed by separate routines (GMBMT, GMBM2) on some systems.

GMBZMD - Set Z Axis

CALL GMBZMD (MODEZ), where

MODEZ = 8* Axis Type + Line Type, with

Axis Type =

- Ø - Normal (Narrow Beam)
- 1 - Defocus (Wide Beam)
- 2 - Write-Thru (Temporary)

Line Type =

- Ø - Normal (No Dash)
- 1 - Dotted

2 - Dot Dash

3 - Short Dash

4 - Long Dash.

Buffered Screen Routines (GMBS--)

GMBSDR - Draw Line Segment to Point

CALL GMBSDR (IX, IY), where

IX = X screen coordinate

IY = Y screen coordinate.

GMBSHR - Draw Hourglass Figure

CALL GMBSHR (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

GMBSPL - Draw Plus (+) Symbol

CALL GMBSPL (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

GMBSXI - Draw X Symbol (XIT)

CALL GMBSXI (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

Buffered Virtual Routines (GMBV--)

GMBVDR - Draw Line Segment to Point

CALL GMBVDR (X, Y), where

X = X virtual coordinate

Y = Y virtual coordinate.

GMBVHR - Draw Hourglass Figure

CALL GMBVHR (X, Y, ISIZ), where

X, Y = Virtual coordinate of center location

ISIZ = Size in screen units.

GMBVNU - Write a Number at a Point

CALL GMBVNU (X, Y, NUM), where

X, Y = Virtual coordinates of start location

NUM = Number to be written.

GMBVPL - Draw Plus (+) Symbol

CALL GMBVPL (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in screen units.

GMBVXI - Draw X Symbol (XIT)

CALL GMBVXI (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in screen units.

Unbuffered Control Routines (GMC---)

GMCAMD - Set Alpha Mode

CALL GMCAMD

GMCBEL - Ring Terminal Bell

CALL GMCBEL

GMCCSZ - Set Character Size

CALL GMCCSZ (ISIZ), where

ISIZ = Size of characters

1 = Largest

2 = Next largest

3 = Next smallest

4 = Smallest.

GMCERA - Erase Terminal Screen

CALL GMCERA

GMCFIN - Finish Use of GRFMAN Routines

CALL GMCFIN (IX, IY, ISIZ), where

IX, IY = Screen coordinates of final position of cursor

ISIZ = Size of character upon termination.

NOTE: Should be last GRFMAN routine to be executed.

GMCGMD - Set Graphics Mode

CALL GMCGMD

GMCHCP - Initiate Hardcopy of Terminal Screen

CALL GMCHCP

GMCINI - Initiate GRFMAN Routines

CALL GMCINI (ISIZ, IIFC, IOFC), where

ISIZ = Character size selection

IIFC = Input File code (terminal)

IOFC = Output File code (terminal)

NOTE: Must be the first GRFMAN routine to execute. Also sets the initial screen and virtual windows. These may be reset by GMSWND and GMVWND. On some systems, will initiate a second task to perform buffered I/O.

GMCNUL - Move Cursor to Next Line

CALL GMCNUL (LINES), where

LINES = Number of lines remaining in current page.

NOTE: Will reset to top of page when full.

GMCZMD - Set Z Axis

CALL GMCZMD (MODEZ), where

MODEZ = 8*Axis Type + Line Type, with

Axis Type =

Ø - Normal (Narrow Beam)

1 - Defocus (Wide Beam)

2 - Write-Thru (Temporary)

Line Type =

Ø - Normal (No Dash)

1 - Dotted

2 - Dot Dash

3 - Short Dash

4 - Long Dash

Unbuffered Screen Routines (GMS---)

GMSBOX - Draw a Box (square)

CALL GMSBOX (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

GMSCLP - Clip a Line Segment at Window Limits

CALL GMSCLP (IX1, IY1, IX2, IY2, ICOD), where

IX1, IY1 = Screen coordinates of point one

IX2, IY2 = Screen coordinates of point two

ICOD = Clip code of point one

NOTE: Point two is assumed to be inside the screen window, while point one is assumed outside. The window limits must be set by a call to GMSWND and can be obtained using GMSLIM.

GMSCOD - Calculate Clipping Code for a Point

CALL GMSCOD (IX, IY, ICOD), where

IX, IY = Screen coordinates of point

ICOD = Clipping code returned

Ø - Inside window

+1 - If less than Y minimum

+2 - If greater than Y maximum

+4 - If less than X minimum

+8 - If greater than X maximum.

NOTE: The code assigned is relative to the screen window as defined by the last call to GMSWND.

GMSCSR - Graphic Cursor Input

CALL GMSCSR (ICHAR, IX, IY), where

ICHAR = Cursor control character

IX, IY = Screen coordinates of cursor location.

GMSCVT - Coordinate Conversion

CALL GMSCVT (IX, IY, X, Y), where

IX, IY = Screen coordinates

X, Y = Virtual coordinates

NOTE: This conversion is based on a mapping from the screen window (as set by the last call to GMSWND) to the virtual window (as set by GMVWND).

GMSDRW - Draw Line Segment to a Point

CALL GMSDRW (IX, IY), where

IX, IY = Screen coordinates of point to which segment is drawn.

GMSDSH - Draw Dashed Segment to a Point

CALL GMSDSH (IX, IY, LEN), where

IX, IY = Screen coordinates as for GMSDRW

LEN = Line type as in GMBZMD.

GMSFRM - Draw Frame at Screen Window

CALL GMSFRM (LEN), where

LEN specifies a line type as in GMBZMD

GMSHRG - Draw Hourglass Figure at a Point

CALL GMSHRG (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units

GMSLIM - Return Screen Window Limits

CALL GMSLIM (IX1, IY1, IX2, IY2), where

IX1, IY1 = Screen coordinates of window minimum

IX2, IY2 = Screen coordinates of window maximum.

GMSMOV - Move Cursor to a Point

CALL GMSMOV (IX, IY), where

IX, IY = Screen coordinates of point

NOTE: Same as GMSDRW, but with a dark (invisible) segment.

GMSNUM - Write a Number at a Point

CALL GMSNUM (IX, IY, NUM), where

IX, IY = Screen coordinates of start location

NUM = Number to be written.

GMSPLS - Draw Plus (+) Symbol at a Point

CALL GMSPLS (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

GMSWND - Set Screen Window Limits

CALL GMSWND (IX1, IY1, IX2, IY2), where

IX1, IY1 = Screen coordinates of window minimum

IX2, IY2 = Screen coordinates of window maximum.

GMSXIT - Draw X at Point

CALL GMSXIT (IX, IY, ISIZ), where

IX, IY = Screen coordinates of center location

ISIZ = Size in screen units.

Unbuffered Virtual Routines (GMV---)

GMVBOX - Draw a Box (square)

CALL GMVBOX (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in virtual units.

GMVCLP - Clip a Line Segment at Window Limits

CALL GMVCLP (X1, Y1, X2, Y2, ICOD), where

X1, Y1 = Virtual coordinates of point one

X2, Y2 = Virtual coordinates of point two

ICOD = Clip code of point one

NOTE: Point two is assumed to be inside the virtual window, while point one is assumed outside. The window limits must be set by a call to GMVWND and can be obtained using GMVLIM.

GMVCOD - Calculate Clipping Code for a Point

CALL GMVCOD (X, Y, ICOD), where

X, Y = Virtual coordinates of point

ICOD = Clipping code returned

- Ø - Inside window
- +1 - If less than Y minimum
- +2 - If greater than Y maximum
- +4 - If less than X minimum
- +8 - If greater than X maximum.

NOTE: The code assigned is relative to the virtual window as defined by the last call to GMVWND.

GMVCSR - Graphic Cursor Input

CALL GMVCSR (ICHAR, X, Y), where

ICHAR = Cursor control character

Y, Y = Virtual coordinates of cursor location.

GMVCVT - Coordinate Conversion

CALL GMVCVT (X, Y, IX, IY), where

X, Y = Virtual coordinates

IX, IY = Screen coordinates.

NOTE: This conversion is based on a mapping from the virtual window (as set by the last call to GMVWND) to the screen window (as set by GMSWND).

GMVDRW - Draw Line Segment to a Point

CALL GMVDRW (X, Y), where

X, Y = Virtual coordinates of point to which segment is drawn.

GMVDSH - Draw Dashed Segment to a Point

CALL GMVDSH (X, Y, LEN), where

X, Y = Virtual coordinates as for GMVDRW

LEN = Line type as in GMBZMD.

GMVFRM - Draw Frame at Virtual Window

CALL GMVFRM (LEN), where

LEN specifies a line type as in GMBZMD.

GMVHRG - Draw Hourglass Figure at a Point

CALL GMVHRG (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in screen units.

GMVLIM - Return Virtual Window Limits

CALL GMVLIM (X1, Y1, X2, Y2), where

X1, Y1 = Virtual coordinates of window minimum

X2, Y2 = Virtual coordinates of window maximum.

GMVMOV - Move Cursor to a Point

CALL GMVMOV (X, Y), where

Y, Y = Virtual coordinates of point

NOTE: Same as GMVDRW, but with a dark (invisible) segment.

GMVNUM - Write a Number at a Point

CALL GMVNUM (X, Y, NUM), where

X, Y = Virtual coordinates of start location

NUM = Number to be written.

GMVPLS - Draw Plus (+) Symbol at a Point

CALL GMVPLS (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in screen units.

GMVWND - Set Virtual Window Limits

CALL GMVWND (X1, Y1, X2, Y2), where

X1, Y1 = Virtual coordinates of window minimum

X2, Y2 = Virtual coordinates of window maximum.

GMVXIT - Draw X at Point

CALL GMVXIT (X, Y, ISIZ), where

X, Y = Virtual coordinates of center location

ISIZ = Size in screen units.

Plotter Routines (GMP---)

GMPDRW - Draw Line to a Point

CALL GMPDRW (X, Y), where

X, Y = Plotter coordinates of the destination.

GMPFIN - Finish Plot Commands

CALL GMPFIN (XSIZE, YSIZE), where

XSIZE = Maximum x-dimension of plot

YSIZE = Maximum y-dimension of plot.

NOTE: Should be the last GRFMAN routine to be executed.

GMPINI - Initiate Plotting

CALL GMPINI (XSIZE, YSIZE, IOFC), where

XSIZE = Maximum x-dimension of plot

YSIZE = Maximum y-dimension of plot

IOFC = Plot output file code.

NOTE: Must be the first GRFMAN plot routine to execute.

GMPMOV - Move to a Point

CALL GMPMOV (X, Y), where

X, Y = Coordinates of the destination.

NOTE: The pen is lifted, and no line is drawn.

GMPPLS - Draw Plus Symbol (+)

CALL GMPPLS (X, Y, SIZE), where

X, Y = Location of center of symbol

SIZE = Total size in x and y of symbol.

GMPXIT - Draw X Symbol

CALL GMPXIT (X, Y, SIZE), where

X, Y = Location of center of symbol

SIZE = Total size in x and y of symbol.

SECTION V

KANSAS GEOLOGICAL SURVEY IMPLEMENTATION

V.1 Introduction

The GIMMAP system has been developed at the Kansas Geological Survey as both a research effort to develop tools for geologic research, and as a working production system to meet various cartographic needs of the Survey. Thus, the requisite operational procedures, data sources, and peripheral systems have been obtained or developed to support a small-scale, experimental production effort.

These facets are a necessary part of any operational system, but quite obviously will differ significantly between any two installations. The input sources, file naming schemes, and digitizing operations will necessarily be non-standard in many ways, but such features developed elsewhere will share some common aspects. For this reason, this section presents the operational developments at the Kansas Geological Survey covering (1) the generation and naming of files for the data base, (2) the use of U.S. Geological Survey (USGS) quadrangle maps as source data, (3) an introduction to map projection and unprojection for the two projections used in USGS maps, and (4) a digitizing operator's manual used at the Kansas Geological Survey.

V.2. File Naming and Generation

The application of the GIMMAP system at the Kansas Geological Survey is primarily constrained to the State of Kansas, for which the USGS 7.5′ topographic map series ("quadrangle" maps) is nearly complete. Because these maps are the most complete, high-quality source of geographic information for the State, the decision was made to use this series as the primary source of input data for GIMMAP at the Kansas Geological Survey.

This decision, and the desire to be able to retrieve data for specific counties has led to the definition of a county patch as the fundamental areal unit in GIMMAP. This term refers to the intersection of a county and a USGS quadrangle map area. County patches may thus be joined to create either a quadrangle map file or to create a county map file for Kansas.

All such county patches are treated as separate entities for digitizing and file generation, and thus a unique naming scheme has been designed. This scheme combines the unique two-letter county codes for the State (see Appendices), and an artificial ordering of the quadrangle areas, consisting of a row and column number based on the rectangular matrix of quadrangles covering the State. For GIMMAP, this matrix consists of 26 rows, numbered from the south, and 61 columns, numbered from the west.

Thus, for example, the county patch defined by the intersection of Douglas County and the Lawrence West quadrangle is named by the concatenation of the county code DG with the row (17) and column (55) numbers for the quadrangle. The (unique) result in this case is DG1755. In practice, the data retrieved from the quadrangle map may be stored in a different set of files than other sources of data, such as digitized geologic contacts. A letter indicating the type of data (such as B for base data) is usually appended to the county-row-column designator to produce, in this instance, DG1755B. This is the base name which is used to name all files connected with this county patch.

In the generation of the base files for any county patch or other area of interest, a complete set includes (generically) the Arc, Node, Point, and Qplot (display) files. In addition, if areas such as geologic zones are to be constructed, the base files would include Border, Label, and Zone files. These files are indicated by the unique first letters of the generic names: A, N, P, Q and B, L,and Z. Appended to the base name, these letters indicate the specific generic names for any set of map base files.

One other significant attribute which may be advantageously incorporated in the naming scheme arises from the need to provide a backup or archival system for the data base. Thus, two sets of base files may be maintained for a particular county patch to protect the data from operator error or other catastrophic events. At the Kansas Geological Survey, such a backup is indicated by the addition of the letter "S" (for Save) to the set of files reserved for this purpose. Similarly, the letter "W" (Working) is appended to the set which is being created or modified.

As a result, the complete set of base files for the Douglas County-Lawrence West county patch containing the quadrangle map base data would be:

Generic Name		Save File	Working Files
ARC	=	DG1755B_AS	DG1755B_AW
BORDER	=	DG1755B_BS	DG1755B_BW
LABEL	=	DG1755B_LS	DG1755B_LW
NODE	=	DG1755B_NS	DG1755B_NW
POINT	=	DG1755B_PS	DG1755B_PW
QPLOT	=	DG1755B_QS	DG1755B_QW
ZONE	=	DG1755B_ZS	DG1755B_ZW

In addition to the base files described above, there exists a set of digitized data files for each map area. Digitized data files are named by the addition of the letter "D" to the base name. Since several such files usually exist for each base, a sequence number, indicating the order of digitization, is appended to each file name. The first digitized data file for the example would be named DG1755B_D1, and the third would be DG1755B_D3.

The creation of syntactically "clean" data files by the SYNEDT program results in the addition of the letter "C" (clean) to the digitized data file name to differentiate the two types. The syntactically clean version of the second digitized data file in the example would be named DG1755B_D2C.

Finally, it should be noted that although these naming schemes may be used to provide valuable information to the GIMMAP program modules, all GIMMAP program modules rely on a particular generic naming system to access the appropriate base files. This set of generic names may be instituted by utility (or operating system) functions and may be derived from any consistent file naming scheme which is adopted.

V.3 USGS Quadrangle Maps

As was mentioned in the preceding chapter, the primary source of input data for GIMMAP processing at the Kansas Geological Survey is the series of 7.5′ topographic maps ("quadrangle" maps) published by the U.S. Geological Survey. These maps are available through the Publication and Sales Office of the Kansas Geological Survey, or through the National Cartographic Information Center, located in Reston, Virginia. An index map displaying the location and names for all Kansas topographic maps and a booklet describing these maps and their symbology are available from both of these sources.

The 7.5′ quadrangle maps are published at a scale of 1:24,000 (1 inch = approx. 1 mile), using either the Lambert Conformal Conic or the (Modified) Polyconic projection. (A discussion of these projections is contained in the following chapter.) The maps are produced in five colors. Manmade features and names are black; hydrologic features and names are blue; urban areas, road classifications, and U.S. land lines are red; woodland areas are green; and contour lines and values are in brown, with contour intervals varying with the relief of the map. Many maps in Kansas have been recently photo-revised, and revised information is shown in purple.

V.4 Map Projection and Unprojection

Map projection is the transformation tool by which the curved, three-dimensional surface of the Earth may be represented in two dimensions as a map. The information contained on a map must differ from the true surface of the Earth, and any system concerned with the accuracy of maps must take into account the projection of these maps.

As noted previously, the GIMMAP system implemented at the Kansas Geological Survey uses primarily the USGS 7.5′ topographic (quadrangle) maps as a data source. In the State of Kansas, these maps have been projected in either the Lambert Conformal Conic or the Modified Polyconic projection. The GIMMAP system contains software to produce these two projections, and their inverse functions. The inverse of projection is referred to as "unprojection," and is the conversion of digitizer table coordinates associated with the map into latitude and longitude values.

These projection and unprojection functions allow GIMMAP to rectify the map edges to their proper locations, and to fuse existing map data from county patch areas into larger scale maps of counties, groundwater management districts, or other areas. This is possible by unprojecting the data to latitude and longitude, rectifying the edges, and reprojecting the data according to the same or different projection parameters. If these functions were not applied, map areas could not be joined without leaving gaps, and lines common to two areas would not meet properly at their common boundary.

There are many sources of information concerning the projection of maps. Among these are Deetz and Adams (1969), Roblin (1969), and Kellaway (1946). With an eye toward computer applications, Tobler

(1962) developed a parametric classification for map projections. The development of projection and unprojection software for GIMMAP relied heavily upon the work in Mahling (1973), CIA (1974), and upon Richardus and Adler (1972), whose developments of the Polyconic and Lambert Conformal projections were found to be the most complete.

Modified Polyconic and Lambert Conformal Conic Projections

Geometrically, the Modified Polyconic projection is defined to be the collection of projections in which each point is projected to the cone which is tangent to the spheroid (model of the Earth) at the latitude of the point being projected. This produces a set of (infinitely many) cones which when opened become circles of varying radii. These circles are then placed so as to be correctly spaced along the Earth's surface at the central meridian. The development of the projected circles and their relative placement is treated in the sources listed above.

The geometry of the Lambert Conformal Conic projection is simply the projection of all points of interest onto the unique cone which intersects the spheroid twice, at what are called the "standard parallels." These standard parallels minimize distortion if chosen to be 1/6th the distance from the top and bottom of the area to be mapped. Again, the development of this projection may be found in the sources listed above.

Unprojection

The unprojection, or conversion to latitude and longitude from table coordinates, of map data used in the GIMMAP system is quite simple and fast. Essentially, the same technique is used for both the Polyconic and the Lambert Conformal Conic projections. The process begins by making an initial guess of the latitude of the point to be unprojected. This guess is based on the nearly linear relationship between latitude and table coordinates for small areas. This guess and the original coordinates lead to a corresponding longitude value. The latitude and longitude are then projected according to the type of projection used, and the resultant point is compared to the original, resulting in an upper or lower bound for the latitude of the point.

A search width is developed for each projection type by determining the maximum possible deflection of a point from the linear relationship stated above. This search width and the upper or lower bound which was found in the first guess are used to envelope the point between an upper and lower latitude boundary. Successive applications of the linear search method eventually lead to a latitude and longitude value whose projection lies within a predefined tolerance value of the original point. In practice, less than four iterations are required to unproject a point within a tolerance far less than the resolution of the digitizing equipment.

V.5 GIMMAP Digitizing Manual

Introduction

This chapter details the operation of digitizing for the GIMMAP system as installed at the Kansas Geological Survey. Much of the information contained in this chapter duplicates that contained in Chapter II.4 on Data Input, but is couched in the production environment of the KGS. This particular presentation is an example of the operational understanding required by digitizing operators using the GIMMAP system.

The actual operating software which controls the Bendix digitizer at the KGS is the COMP-U-GRID system, provided by Computer Equipment Corporation. This software is quite complex and versatile, but only a minimal subset is actually used in GIMMAP digitizing. This subset (shown below) is very basic, and can be found in some form in nearly all digitizing systems.

GIMMAP DIGITIZING COMMANDS

Command	Use
Dƀ	Enter feature code or flag
Mƀ	Enter map header
ME	Mark end of data file on tape
MF	Move tape forward
MR	Rewind tape
N	Start new data file

GIMMAP Digitizing

GIMMAP is a system for capturing geographic data, and editing, modifying, and producing maps by computer. The data are taken from source maps and are entered into the computer as geographic points represented by a set of x, y, and sometimes z coordinates. These data points (or coordinates) enter the computer by a process known as "digitizing." When a map is digitized, the data on that map are converted into mathematical coordinates on an x-y plane.

GIMMAP is a set of programs or computer instructions which tells the computer how to interpret and manipulate the digitized information to produce a finished product which is itself a map. This new map differs from the original source map, because while it can be plotted on paper like the original, it also exists in the memory of the computer, and in this form can be stored for future use. This increases the value of the map, because it can now be reproduced at any desired scale, and individual features such as roads can be changed without having to draw a new map. The user also can choose to plot only certain types of features, instead of the entire map.

Since GIMMAP is a specialized group of programs, it requires the data to be in a certain format. Thus, there is a specific procedure to be used in digitizing a map for the GIMMAP system. This procedure, or syntax, results in an end product called a "digitized data file."

Digitized Data File

The digitized data file consists of three parts: the map header, the control points, and the feature information.

Map Header

The map header identifies the data file and is used by the computer as a reference when searching the data base; the header is the "name" of the data file. A special command is used to transmit map header data to the computer. The command is the letter M, followed by a blank space. The map header identifies the area being digitized, both by an abbreviation of the county name and by a four-digit code which is explained below. The header also includes other information, described below. Entry of a typical map header is shown in Figure V.5.1, using the conventions which are specific for applications in Kansas.

```
M b̸ D G 1 8 5 6 B - D 2 b̸ P M  b̸ 4 A P R 8 Ø
  \___/ \_____/ \___/   \___/  \___________/
    1      2      4   5   6          7
  \_________/
       3
```

Figure V.5.1. Format for entry of map header (the symbol "b̸" represents a blank space).

The following numbered items correspond to the illustrated map header above.

(1) DG is a county code abbreviation for Douglas County. Each of the 105 counties in Kansas is identified by a unique two-letter county code.

(2) The four-digit number describes the map location on a statewide quadrangle map grid system of 25 rows and 61 columns. The first two digits are the row number, the second two are the column number (in this example, row 18, column 56).

(3) The combination of the county (specified by the two-letter code) and the quadrangle (specified by row and column numbers) represents a unique area called a "county patch" in GIMMAP, and is the basic unit to be digitized.

(4) These letters describe the data form and data source. The first letter (or pair of letters) specifies the source document type. The second letter is separated from the first letter (or letters) by an underscore, and describes the form of the data in the file.

(5) This is a sequential number identifying the work session during which the map was digitized (the second work session in this example).

(6) A blank, followed by the digitizer's initials or name.

(7) Another blank, followed by the date of the work session.

Control Points

The control points are the coordinates of the four corners of the map (or the corners of a circumscribing rectangle, if the map is not rectangular in shape). These define an area which contains data to be digitized, and form the borders of the digitized map. The control points are digitized in counter-clockwise order, starting and ending with the southwest corner of the map (see Fig. V.5.2).

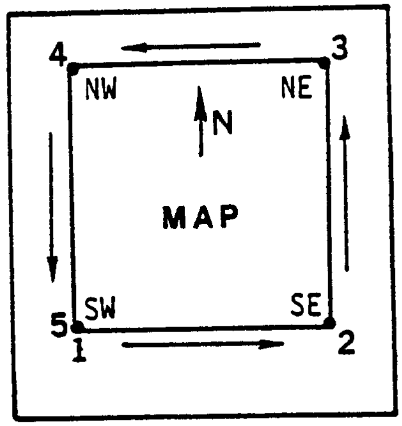


Figure V.5.2. Digitizing sequence for control points.

Feature Information

The feature information is that data represented on the map by lines, dots, or shaded areas. In GIMMAP terminology, lines are called "arcs," dots are called "isolated points," and areas are known as "zones." Some other GIMMAP terms which are important for understanding digitizing are "nodes, "islands," and "bridging."

Nodes are points which have one or both of the following characteristics: (1) they are endpoints of arcs (either the starting or ending point), (2) they are the intersections of two (or more) arcs of the same type. Nodes play a special role in digitizing, and are treated somewhat differently than the other points along an arc.

An island is an arc which forms a complete loop and has no natural starting or ending point. To digitize an island, an arbitrary point is chosen to serve as both the starting and ending points for the arc.

Bridging occurs when two different arcs connect the same two nodes. An arbitrary node must be added somewhere along the interior points of either of the arcs.

Each type of feature on a map is represented by a unique four-digit number called its "feature code." A complete list of feature codes is given in the appendices. The command D, followed by a blank, is used to enter feature code information. Below is an example of the syntax for the feature code describing a two-lane county road, as it would be seen on actual output from the digitizing console:

D␢2254.,Ø.

The feature code header is entered as a pair of real numbers (containing decimal points). The first number is the feature code entered as a real number; the second real number is Ø.

A special marker called a "discontinuity flag" is used to indicate that two consecutively digitized features are not connected, as would otherwise be assumed. The discontinuity flag consists of one pair of negative coordinates and may be entered by digitizing a point with known negative coordinates, or by use of the D␢ command:

D␢ - 1.Ø, - 1.Ø

Another marker, called a "header flag," is used to signal the entry of a feature code header. This flag consists of two consecutive pairs of negative coordinates (two consecutive discontinuity flags), and immediately precedes the feature code header.

Digitizing Preparations

Careful preparation will help reduce errors and increase the digitizer's efficiency. The following suggestions should be considered before beginning to digitize.

1. A high-quality, physically undamaged map should be used as the input (data) source.

2. A complete list of all features to be digitized and their respective feature codes should be compiled, and be readily available.

3. The contents of the map header should be decided and written down. Locate the particular quadrangle on the quadrangle reference map, find its row and column numbers, and look up the county code for the county patch being digitized.

4. Node locations should be decided upon for the arcs to be digitized. These locations can be circled lightly in pencil on the map (unless it is an advance sheet, then a clear plastic overlay must be used).

5. The map should be studied for the occurrence of islands and bridging, and artificial nodes should be chosen to remedy these situations. These also may be marked lightly in pencil.

6. The sequence of digitizing should be decided upon before starting.

7. The map should be firmly secured to the table with tape. Care should be taken to minimize wrinkling and stretching of the map. The layout of the digitizing table is shown in Figure V.5.3.

Taping the Map to the Digitizing Surface

1. The map may be placed with the x-axis in a vertical or horizontal orientation (or any orientation that is convenient).

2. Tape down one corner of the map using an ample piece of masking tape (a 1" to 1.5" strip will do well). Smooth out any bumps or wrinkles in the map by brushing your hand across it, starting from the taped corner and moving towards the diagonally opposite corner. Then, tape this corner down, keeping the map pulled taut. Repeat this proce-

dure for the other two corners of the map, making sure to reduce wrinkles and bumps as much as possible, since they will greatly affect the accuracy of the digitized points.

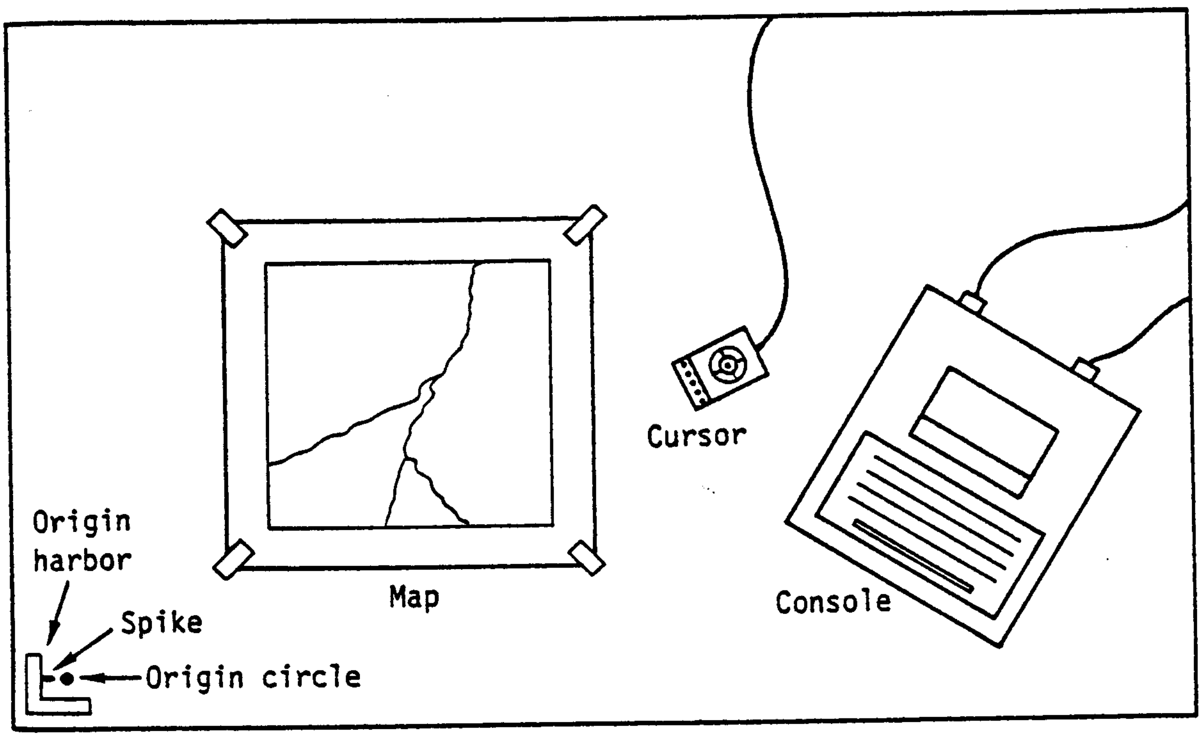


Figure V.5.3. Layout of the digitizing table.

Setting the Digitizing Table Origin

1. Move the cursor so that it is inside the origin harbor so that the metal spike fits into the corresponding notch in the cursor as shown in Figure V.5.4.

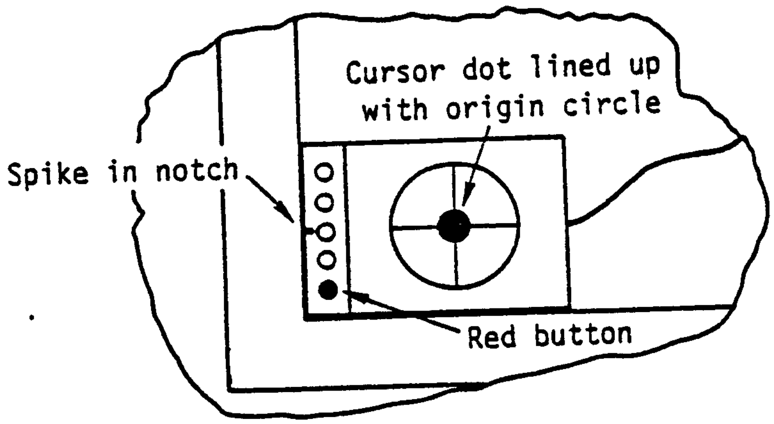


Figure V.5.4. Setting the digitizing table origin.

2. Line up the cross-hair circle and the black origin circle as best as you can. When they are lined up, carefully press the RED cursor button. The RED cursor button is used only for setting the origin.

3. The origin is now set. The map can be placed anywhere within the area defined by the four black circles (dots), one of which is located near each corner of the digitizing table. This area is called the "digitizing area."

Digitizing Theory

The previous suggestions for digitizing preparation will be extremely helpful in planning how to digitize the data represented by the various groups of features. Establish a sequence or order, by which you plan to go from one feature type group to another with maximum efficiency. You must organize the many features on the map into groups of the same feature code (write down these groups if it will help you remember them), then figure out how you wish to go about digitizing these groups. While establishing this order, consider the suggestions below so you will have a good idea of what you are going to do before you start.

1. There are three types of features: isolated points, arcs, and zone-boundary arcs. Digitize all isolated points first, all non-zone-boundary arcs second, and all zone-boundary arcs last.

2. Within one feature type there may be many groups of features, each with its own feature code. Digitize all features of the same code in one group. This will help reduce the number of omissions and also the number of feature code headers used.

3. If there are many features of the same type and feature code, digitize them in an order similar to reading a page in a book (left to right, top to bottom). This will help reduce the number of omissions.

Two suggestions to be considered while digitizing are as follows:

1. Draw a light (but obvious) pencil mark through all features which have been digitized. This should be done at a convenient time, such as after entering a flag.

2. When entering a flag (by digitizing points with negative coordinates with the cursor, or by typing in D␢-1.Ø,-1.Ø at the console) make sure to remember where you stopped digitizing so you can return there after entering the flag.

Point vs. Line Mode

There are two forms of digitizing: point mode and line mode. In point mode, a single point is transmitted at each depression of the cursor button. WE WILL ALWAYS BE DIGITIZING IN POINT MODE! In line mode, a continuous stream of points is transmitted while the cursor button is depressed.

Cursor Contact

THE CURSOR MUST STAY IN CONTACT WITH THE MAP SURFACE AT ALL TIMES during the digitizing process. If you lift the cursor off the board surface, a buzzer will sound, to let you know that the table origin has been lost. Use the following procedure if this happens:

(1) Press the red cursor button to silence the buzzer.

(2) Reset the coordinate axes by:

  (a) resetting the table origin, and

  (b) redefining the x-axis, scale, and reference point for the map.

(3) Start digitizing from the point where the axes were destroyed.

Correcting Typing Errors

If you make a mistake when typing and you catch it before you press the carriage return, press the "CTRL" and "X" keys on the console, which will erase the line. Type in the correct text followed by a carriage return.

Digitizing Accuracy

BE AS ACCURATE AS IS HUMANLY POSSIBLE when digitizing data from maps. The data are going into a statewide database, so accuracy is very important. Use the cursor window crosshairs to get as good a fit as possible. When digitizing nodes, make sure that you do not move the cursor between the first and second pressings of the cursor button. If the cursor is moved, the computer will read it as two single points instead of as a node.

Digitizing Isolated Points

Isolated points such as quarry, mine, or well locations are treated differently than arcs and zone-boundary arcs by GIMMAP programs. All types of isolated points have feature codes from 1000 to 1999, and are digitized once only, one right after the other. Discontinuity flags are not used, because by definition there is always a discontinuity between two isolated points.

Digitizing Operations

In the following exercise we will run through an actual digitizing session, following the "conversation" between the computer and the digitizer, through the computer console. Commands, instructions, and cautions are discussed in the comments column. Some of the symbols refer to this exercise only, and are not used in real operation with the computer. These special symbols are:

b̸	a blank space
Ø	the number zero
O	the letter O
<cr>	carriage return
n	sequence or step number for this exercise.

In the computer/operator column, all computer generated messages and operator responses will be in CAPITAL LETTERS; operator responses will be UNDERLINED.

Seq. No.	Computer/Operator	Comments
1		Boot up the Nova. See "Booting Up the Nova" in appendices.
2		Set the digitizing table origin. See "Setting Cursor Origin" in appendices.

Seq. No.	Computer/Operator	Comments
3		Set up desired map on the digitizing table. See instructions in appendices.
4	?	Computer requests program name code for the CEC digitizing program.
	CECASCII <cr>	Response.
5	CEC DIGITIZATION SYSTEM DIG. 2 POINTS ALONG X-AXIS	Digitize two points along the x-axis of the map. For GIMMAP applications, digitize the southwest corner, then the southeast corner of the map. Digitize these AS ACCURATELY AS POSSIBLE, as they will affect the accuracy of the digitized coordinates.

Seq. No.	Computer/Operator	Comments
6	SUPPLY SCALE <cr>	Requests scale information for the map. By entering a carriage return, we tell the computer that a one-to-one relationship exists in this case. This is true for all GIMMAP digitizing.
7	DIG. REFERENCE POINT	Digitize the southwest corner of the map. Again, digitize as accurately as possible.
8	SUPPLY X,Y VALUES AT REF. PT. 1.Ø,1.Ø <cr>	Entered as a pair of real numbers (requiring a decimal point) in the form of coordinates. All points farther than one unit southwest of this point will have negative coordinates, and may be used as flags.

Seq. No.	Computer/Operator	Comments
9	READY	The coordinate plane has been initialized. Dismount boot tape, then load digitizing tape. If the digitizing tape is new, we may start in at the beginning of the tape; but if it has been used, we must advance the tape past all previously recorded data files. (If using a new tape, skip Step 10.)
10	MF2ØØØ <cr>	In order to advance the tape past the previous files, we use the MF command, which advances the tape the specified number of records (in this case, 2000) or until the filemark is reached. Any number of records may be specified, but the tape will advance past only one filemark at a time.

Seq. No.	Computer/Operator	Comments
11	TAPE STATUS:1ØØØ1ØØ1Ø1 READY	Indicates that the filemark has been reached (if we have advanced past all previous data files, we may now start to digitize). The computer is ready to continue.
12	RX <cr> READY	This command turns off the echoprint feature of the CEC digitizing system.
13	Ob̸4 <cr> READY	This command instructs the computer to record all following information and data on the digitizing tape.
14	Mb̸DG1865B-D2b̸PMb̸4APR8Ø <cr> READY	All information following the "Mb̸" is the map header for the county patch being digitized. The M is the command used to enter the header into the computer.

Seq. No.	Computer/Operator	Comments
15		Digitize control points, starting with the south-west corner of the map and continuing around the map in a counter-clockwise fashion, ending with the southwest corner of the map. Accuracy is important!
16	Db̸4330.,Ø <cr>	All information following "Db̸" is the feature code header. The feature code header is entered using the D command followed by a blank. The feature code is entered as a pair of real numbers, like coordinates, where the x-coordinate is the feature code and the y-coordinate is Ø. (zero). A complete list of feature codes is given in the appendices.

Seq. No.	Computer/Operator	Comments
17		Digitize all arcs of this feature code, remembering to use discontinuity flags between unconnected arcs of the same type. When finished digitizing all arcs of the same type, enter a feature code flag (two discontinuity flags) to denote the end of this feature code data set.
18	D𝖻111Ø.,Ø. <cr>	Digitize all arcs of this feature code type. In this case, though, we are dealing with isolated points, so we need not use a discontinuity flag between unconnected arcs because, by definition, we have no connected arcs.

Seq. No.	Computer/Operator	Comments
19		When all arcs for this feature code have been digitized, enter a feature code flag, then enter the next feature code. Continue in this manner until all feature code groups have been digitized, thus completely digitizing the county patch. The end of this file must be marked with a filemark.
20	ME <cr>	Filemark command issues a filemark to denote the end of this data file.
21	TAPE STATUS:1ØØØ1ØØ1ØØØ1	Indicates the filemark has been issued.

Seq. No.	Computer/Operator	Comments
22		The digitizer operator now has two options: (1) continue with work session by digitizing another county patch, or (2) end the work session by rewinding the tape and shutting off the system.
23a	N <cr>	OPTION 1: CONTINUE WORK SESSION. "New Map" command; go to Step 5.
23b	READY	OPTION 2: CONCLUDE WORK SESSION. Follow remaining steps.

Seq. No.	Computer/Operator	Comments
24	MR <cr>	Rewind command: Rewinds digitizing tape to the LOAD point. Next press RESET and then REWIND buttons on the front panel of the tape drive to completely rewind tape. Dismount tape and place it in its storage case.
25		After dismounting the tape, press the power button on the tape drive to turn it off. Next, turn the power key for the Nova to the OFF position. Last, flip the power switch to the console to the OFF position.
26		Fill out Nova usage and project completion records. Label the digitizing tape to show what has been added.

GIMMAP Digitizing Terminology

Arc	Line (a set of points) between two nodes.
Bridging	Situation where two (or more) arcs have the same starting and ending nodes, so distinguishing between them can be a dilemma. This is solved by using an "artificial node" along the interior points of either arc, to preserve the uniqueness of the arcs.
Continue Node	Special-purpose node. Digitized TWICE. Used where the endpoint of an arc of the same type intersects the arc being digitized but we do not wish to end the arc at that point. Used to mark this endpoint along the interior of the first arc (such as a branch in a river).
County Patch	Intersection of the county to be digitized and the quadrangle map boundaries; that is, the portion of the county that is being digitized that is represented on the quad map being used.

Discontinuity Flag	A pair of coordinates with negative X, Y values entered by one of two methods: (a) digitize a point to the southwest of the reference point of the map, or (b) use the D command, followed by -1.Ø, -1.Ø (Db-1.Ø,-1.Ø). Indicates to the computer that another, but unconnected, arc of the same type is going to be digitized.
End Node	Last point along an arc. Digitized TWICE.
Feature Code	A four-digit integer which uniquely describes any map feature (such as roads, county boundaries, etc.). See the appendices for a complete listing.
Feature Code Header	Feature code which describes the arc or group of arcs being digitized. Entered as a set of X, Y coordinates.
Entry Command:	Db
Format:	111Ø.,Ø.
Header Flag	Two consecutive pairs of negative X, Y coordinates entered by (a) digitizing a point to the southwest of the reference

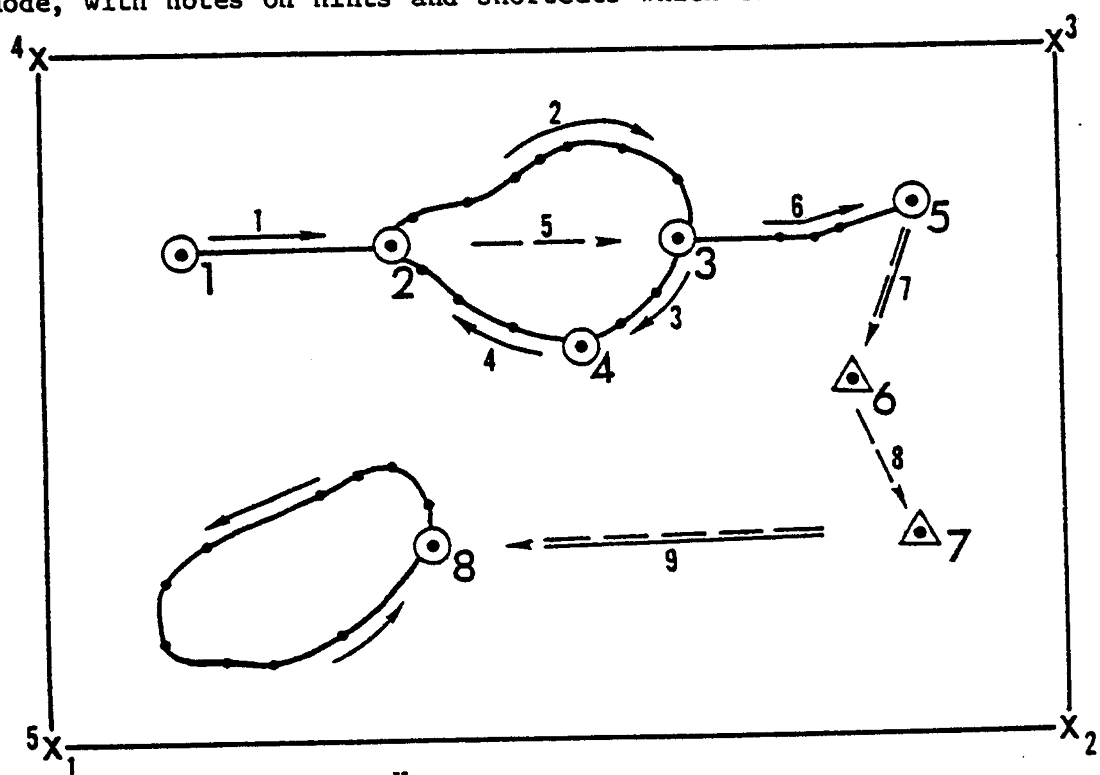
point twice, or (b) by entering the following twice: DØ-1.Ø,-1.Ø. Indicates that the next feature to be digitized will be of a different code.

Interior Points	Points along an arc which help to define its shape. Digitized once only.
Island	An arc which forms a complete loop and has no natural starting or ending point. In this case we choose an arbitrary point to serve as an "artificial" node which acts as both the start and end nodes.
Isolated Point	An arc with no interior points. Digitized once.
Map Header	Textual information which describes the map being digitized. Identifies the county patch, data source, digitizing session, digitizer's name, and date digitized.
Entry Command:	MØ
Format:	DG1856B-DØPMØ4APR8Ø

Node	Point which is (a) an endpoint of an arc, or (b) the intersection of two (or more) arcs of the same type. NODES ARE DIGITIZED TWICE. There are three types of nodes: start node, end node, and continue node.
Quad	USGS quadrangle map. Source for digitizing data.
Quad Reference Map	A map of Kansas which is broken down into a grid of 26 rows (horizontal) and 61 columns (vertical) representing the quad maps for this State. Enables us to create reference numbers for map headers when digitizing.
Start Node	First point along an arc. Digitized TWICE.
Zone	A continuous area on a map characterized by some common feature and therefore shaded or colored, such as a forest, oilfield, soil unit, or geologic formation. This area is enclosed by one or more arcs.

Digitizing Example Problem

The diagram below represents a map to be digitized. Nodes, interior points, control points and special cases such as isolated points and bridging are all used in this example, to give you some experience in the thought process that goes into digitizing a map. We will work through this process from digitizing the control points to the last end node, with notes on hints and shortcuts which can be taken.



Key

- ⊙n — Node (n = sequence no.)
- △n — Isolated Point
- • — Interior Point
- Xn — Control Point
- —n→ — Digitizing Sequence
- –n–→ — Discontinuity Flag
- =n=→ — Feature Code Flag (change in feature code)

Let us assume that the map has already been "set-up" using the CEC Digitizing Program, the map header has been entered, and that we are now ready to digitize the control points.

Starting with the southwest corner, digitize the control points in counter-clockwise order by following sequence numbers 1 through 5. The southwest corner is also the last control point to be digitized. When digitizing control points, you need only press the cursor button once for each point. This differs from nodes, where the button is pressed twice at each point.

We can now start to digitize the various features on the map. First we enter the feature code header for the arcs (by using the DØ command), connecting nodes 1, 2, 3, 4, 5. Next, Node 1 (the start node) is digitized by lining up the point in the cursor window with the point on the map, then pressing the cursor button twice (being careful not to move the cursor between the first and second pressings).

Since the arc connecting nodes 1 and 2 is a straight line, we need not digitize any interior points, and we can next digitize node 2 (nodes are pressed twice, while interior points, control points, and isolated points are pressed just once). After digitizing node 2 (a continue node) we move right along towards node 3, digitizing an appropriate number of interior points (enough to get a close approximation of the line represented) until we reach node 3, which in turn is digitized (node 3 is also a continue node). We now move towards node 4, taking interior points where necessary.

Node 4 is called an "artificial" node, because it is digitized to help prevent a situation known as "bridging," which occurs when two nodes are connected by two (or more) arcs. If node 4 were not introduced, nodes 2 and 3 would be connected by two arcs which could not be uniquely described.

Node 4 is digitized then, and we move towards node 2 to complete the loop. We digitize an appropriate number of the interior points, then finish the arc by digitizing node 2 again, this time as an end node. We now must use a discontinuity flag to indicate that the next arc does not connect nodes 2 and 3. This is done by digitizing a point to the southwest of the southwest corner of the map (the reference point), or by using the alternate method. The alternate method of entering a flag is to type in a pair of negative coordinates -1., -1. by using the D command. The line typed in will look like this: D$\cancel{b}$-1.,-1. After the discontinuity flag has been entered, we digitize node 3, this time as a start node for the arc connecting nodes 3 and 5. We then move along the arc towards node 5 (taking interior points when necessary). We finish this arc by digitizing node 5 (remember, for nodes the button is pressed twice!). When node 5 is digitized, we have finished digitizing all the arcs of that feature code, so we must use a header flag, which is two successive discontinuity flags.

The next feature is an isolated point (sequence no. 6). We enter the feature code header, identifying it as an isolated point and what it represents. Isolated point 7 in this example is also of the same feature code. There is a discontinuity between point 6 and point 7; that is, they are not connected by an arc. In this situation the feature code header serves as an indication of the discontinuity, so we don't enter a discontinuity flag between them. With this knowledge in mind we digitize point 6 (pressing once) and then digitize point 7 (pressing once) following these points with a header flag.

After the header flag is entered, we enter the feature code header for the next arc. In this case the next arc is an island, where no node is defined by arc intersection (or termination). Again, we create an artificial node (node 8) and use it as both the start node and the end node for this arc, thus forming the complete loop. Node 8 is digitized as a start node (pressing twice) then we move around the loop digitizing interior points when needed. We complete the loop by again digitizing node 8, this time as the end node (pressing twice). We have now finished digitizing the map, so a header flag is entered to signify the end of the data file.

The last remaining step is to enter the "ME" command which issues a filemark for the data file on the magnetic tape. We can now do one of two things: (a) digitize another map (county patch), or (b) shut down the system. Procedures for both (a) and (b) are outlined in the Digitizing Operations section of this manual.

SECTION VI

CONCLUSIONS

Introduction

This chapter reiterates the goals and achievements of the development of the GIMMAP system at the Kansas Geological Survey, states the basic conclusions of this development, and outlines further research which is currently being considered.

Goals

The GIMMAP system has been developed to fulfill the cartographic needs of the Kansas Geological Survey within the existing, limited minicomputer facility. As previously stated (Chapter I.1), the specific goals of this development included the basic GDP functions of (1) accurate data capture, (2) generation and support of a cartographic database, (3) graphical display of data, (4) interactive graphical editing, (5) attribute and graphical data addition, (6) selective retrieval based on location and attribute values, (7) production of intermediate plots for edit-completion testing, (8) production of high-quality, color-separated scribecoats for multicolor map printing, and (9) retrieval of "clean" map data for transfer or archival operations.

Achievements

The GIMMAP system is a comprehensive GDP system consisting of a detailed digitizing operation; eleven modular programs which create, interactively edit, update and selectively retrieve data from a complex cartographic database consisting of seven random-access, inter-connected base files; two utility libraries which support random file management and provide graphical functions for an interactive display terminal and a flatbed plotter; and considerable documentation in the form of a digitizing manual, and a system manual for operations and programming considerations.

Applications of the facilities in GIMMAP have resulted in the production of a color-proof for the experimental geologic map of the Lawrence East quadrangle and "The Geology of Lawrence West, Kansas," published as Map M-14 by the Kansas Geological Survey. A further application is the continuing creation and maintenance of the Kansas Database, a statewide cartographic database which has been used to produce a base map of (Kansas) Groundwater Management District No. 4 for use as a research tool by the Geohydrology Section of the Kansas Geological Survey.

A review of the GIMMAP system was presented by the author at the Fourth International Symposium on Computer-Assisted Cartography (AUTO-CARTO IV), held in Reston, Viriginia, in November, 1979. The presentation has since appeared in the proceedings of that symposium (Ross, 1980). Furthermore, much of the material in this thesis will appear in the "GIMMAP User Manual," to be published by the Kansas Geological Survey as No. 9 in the Series on Spatial Analysis.

Conclusions

The Geodata Interactive Management Map Analysis and Production system, GIMMAP, has been successfully implemented on the Data General Nova and Eclipse minicomputers at the Kansas Geological Survey. The GIMMAP system fulfills the goals outlined above, and has been developed in approximately three years by a minimal staff and for a very modest cost.

The construction and maintenance of the Kansas Database via the GIMMAP system will support the production of base maps incorporating other data appended by researchers at the Kansas Geological Survey. The Kansas Database will provide the data for the creation of maps for any selected area in the State, produced at any desired scale, and containing any selected subset of map features.

Further Research

A principal area of further research will be the development of software to provide line and feature generalization, and inter-map interactive editing facilites. These, in turn, would support the creation of higher order (county, state, or other area) base files through the concatenation or "quilting" of new data in the quadrangle base files. The new base files can then be used to facilitate data input from existing county and state maps, and production of county, state and other special-area maps. Data entered from existing county and state maps could be filtered down to the lower level base files to create "first approximation" maps which could then be edited in the field.

A second major research area is the development of a subsystem to incorporate additional (geographical) attribute data into the GIMMAP system. Thus, for example, a geohydrologist may wish to connect depths, pollution measurements and mineral contents to river features; a geographer might want population counts, area, government type, and economic information associated with the graphical features representing cities; a seismologist might want to attach severity levels, lengths of duration, and damage information to the locations of epicenters of earthquakes; and a surface geologist may want to connect thickness, dip, strike, and fault information to the contact and formation data already available.

These attribute-to-graphical data connections could be supported by the development of a relational database management system for attribute data and a corresponding graphical, query language such as that suggested by Zloof (1977). If such a system were properly designed and interfaced to the existing GIMMAP system, an unsophisticated user could enter, append, update, and retrieve attribute data associated with the graphical features of a map, while viewing these features on a graphics display terminal.

Furthermore, analysis and retrieval operators could be incorporated in the graphical query language to provide automatic retrieval of the graphical data, based on operator-selected attribute values and values of functions applied to multiple attributes combined with geographic location criteria. Thus, the unsophisticated operator could produce maps through the application of simple relational operators on attribute data presented to the operator in simple tabular form.

APPENDICES

A.1 RAFMAN Error Codes

Error Number	Meaning
1	Attempt to RMPUTR record at or before accounting record.
2	Attempt to RMPUTR record beyond MAX record number.
3	Attempt to RMPUTR record which is in the freelist.
4	Attempt to RMGETR record not in the freelist.
5	Attempt to RMGETR record beyond MAX record number.
6	Attempt to access freelist record as a used record.
7	End-of-File before MAX record.
8	Attempt to access record at or before accounting record.
9	Attempt to access record beyond MAX record number.
10	Illegal degree in Node record.
11	Inconsistency between Node and Arc files.
12	Inconsistency within Node file.

A.2 XYNETICS Plotter Commands

A) The XYNETICS plotter accepts 3-word records written in binary format:

WRITEȸBINARYȸ(NPLOT)X,Y,IPEN

where:

NPLOT = file code of the plot file

X,Y = real-valued coordinates

IPEN = integer value

Action	X	Y	IPEN
1 START PLOTTING (first record on file, to initialize plotting software)	5.0	0.0	0
2 GIVE OVERALL PLOTTING AREA DIMENSIONS (multiplotting) PPSZX > 57" and PPSZY < 42"	PPSZX	PPSZY	-4
3 START A PLOT PLSZX,PLSZY = plot sizes	PLSZX	PLSZY	-2

4	MOVE PEN X,Y = location to which pen is moved	X	Y	3 pen up 2 pen down
5	SELECT OR CHANGE PEN X,Y is irrelevant; select one of four pens	X	Y	-11 pen 1 -12 pen 2 -13 pen 3 -14 pen 4
6	END A PLOT	PLSZX	PLSZY	-3
7	SEPARATE 2 PLOTS (multiplotting) SDISTX,SDISTY = separating distances between plots	SDISTX	SDISTY	-5
8	END ALL PLOTTING	0.0	0.0	-1

B) MULTIPLOTTING

The XYNETICS works columnwise. It sets plots sequentially in columns starting the first plot at the lower left corner of the plotting area. The second plot is set in the increasing y-direction and so on. If there is no place for a plot in a column, the system starts another column at SDISTX inches of the largest plot of the previous column.

A.3 File Code Assignments

All programs in the GIMMAP system access the base files using the following file code assignments.

File Code	File Name	Physical Record Length (Words)
1	ARC	18
2	POINT	33
3	NODE	18
4	QPLOT	17
5 or 11	Terminal Input	—
6 or 10	Terminal Output	—
7	ZONE	13
8	BORDER	7
9	LABEL	16
>11	Other Data Files	—

A.4 Feature Codes

***** POINTS ***** 1000 +

11--	** Control, Recreation, or Interest **
10	Section Point
20	Horizontal/Vertical Control
30	Point of Interest
40	Historical Marker
50	Recreation
1	Golf Course
12--	** Cities, Towns **
10	0 - 2,500 (Population)
20	2,500 - 5,000
30	5,000 - 25,000
40	25,000 - 100,000
50	100,000 - up
-0	Incorporated
-1	Unincorporated
13--	** Wells, Springs **
10	Water
20	Oil

30	Gas
40	Spring
-0	Producing
-1	Abandoned

14--	** Pits, Quarries, Mines **
10	Open Pit
20	Mine
30	Quarry
40	Prospect
50	Shaft or Tunnel Entrance
-0	Producing
-1	Abandoned

15--	** Buildings **
10	Dwelling or Employment
20	Church
30	Cemetery
40	Tower
1	Radio
2	Transmission Line
3	Water
4	Lookout
50	Substation (Power)

***** LINES ***** 2000 +

21— ** Borders, Boundaries **

10 U.S. Survey Lines

1 Township

2 Range

3 Section

20 Quadrangle Map Edge

30 County Line

40 State Line

50 City Boundary

60 Airport

70 Park

1 National

2 State

3 County

4 City

80 Cemetery

22— ** Highways, Roads **

10 Toll Roads

20 Divided Highways

30 2-Lane Highways

40 More than 2-Lane Streets

50 2-Lane Streets

60	Unimproved Roads
70	Proposed Roads (or under construction)
80	Bridge or Tunnel
-1	Interstate
-2	U.S. or Federal
-3	State
-4	County

23--	** Railroads, Airport Runways **
10	Single Track
20	Multiple Track
30	Railway Bridge or Tunnel
40	Airport Runway

24--	** Hydrology (Lineal) **
10	Perennial Stream
20	Intermittent Stream
30	Canal
40	Dam
50	Dike or Levee

25--	** Transmission Lines **
10	Power
20	Pipeline

1 Gas

2 Oil

3 Water

4 Other

30 Telephone

40 Substation (Power)

26-- ** Intra-formation Geology **

10 Visible Fault

20 Approximate Fault

***** LINES ***** 3000 +

3--- ** Topographic Contours **

ABC Elevation in Feet

A-- Thousands

-B- Hundreds

--C Tens

***** AREAS ***** 4000 +

41-- ** Hydrology (Areal) **

10 Perennial Streams

20 Intermittent Streams

30 Permanent Lakes

40 Intermittent Lakes

50 Reservoirs

42-- ** Inter-formation Geology **

10 Actual Contact

20 Approximate Contact

30 Actual Fault

40 Approximate Fault

50 Underwater Geology

1 Actual Contact

2 Approximate Contact

3 Actual Fault

4 Approximate Fault

43-- ** Other Boundaries **

10 State Line

20 County Line

30 Quadrangle Map Edge

***** EDIT ***** 10000 +

1xxxx ** Added or Re-digitized Features **

xxxx Normal Feature Code as Listed Above

A.5 Kansas County Codes

Code	County
AL	Allan
AN	Anderson
AT	Atchison
BA	Barber
BT	Barton
BB	Bourbon
BR	Brown
BU	Butler
CS	Chase
CQ	Chautauqua
CK	Cherokee
CN	Cheyenne
CA	Clark
CY	Clay
CD	Cloud
CF	Coffey
CM	Comanche
CL	Cowley
CR	Crawford
DC	Decatur
DK	Dickinson
DP	Doniphan
DG	Douglas
ED	Edwards
EK	Elk
EL	Ellis
EW	Ellsworth
FI	Finney
FO	Ford
FR	Franklin
GE	Geary
GO	Gove
GH	Graham
GT	Grant
GY	Gray
GL	Greeley
GW	Greenwood
HM	Hamilton
HP	Harper
HV	Harvey
HS	Haskell
HG	Hodgeman
JA	Jackson
JF	Jefferson
JW	Jewell
JO	Johnson
KE	Kearny
KM	Kingman
KW	Kiowa
LA	Labette
LE	Lane
LV	Leavenworth
LC	Lincoln
LN	Linn
LG	Logan
LY	Lyon
MN	Marion
MS	Marshall
MP	McPherson
ME	Meade
MI	Miami
MC	Mitchell
MG	Montgomery
MR	Morris
MT	Morton
NM	Nemaha
NO	Neosho
NS	Ness
NT	Norton
OS	Osage
OB	Osborne
OT	Ottawa
PN	Pawnee
PL	Phillips
PT	Pottawatomie
PR	Pratt
RA	Rawlins
RN	Reno
RP	Republic
RC	Rice
RL	Riley
RO	Rooks
RH	Rush
RS	Russell
SA	Saline
SC	Scott
SG	Sedgwick
SW	Seward
SN	Shawnee
SD	Sheridan
SH	Sherman
SM	Smith
SF	Stafford
ST	Stanton
SV	Stevens
SU	Sumner
TH	Thomas
TR	Trego
WB	Wabaunsee
WA	Wallace
WS	Washington
WH	Wichita
WL	Wilson
WO	Woodson
WY	Wyandotte

BIBLIOGRAPHY

Berman, R., and Stonebraker, M., (1976), "GEO-QUEL, A System for the Display and Manipulation of Geographic Data," Dept. of E.E. and C.S., Univ. of Calif., Berkeley, CA.

Boyle, A. R., (1979), "Scan Digitization of Cartographic Data," presented at the NATO Advanced Studies Institute on Map Data Processing, June, 1979, in Maratea, Italy, Lecture Notes, 24 p.

Brooks, K., and Pease, J. R., (1978), "Geographic Information Systems: A Review," as part of "Environmental Impact Assessment: A Framework for Local Participation and Decision-Making," Oregon State Univ. Extension Service, 53 p.

Calkins, H. W., and Tomlinson, R. F., (1977), Geographic Information System, Methods and Equipment for Land Use Planning, 1977, for the International Geographical Union, 394 p.

Cao, T. T., (1978), "GIMMAP: An Example of a Geographic Information Management and Mapping System," Proc. Am. Cong. on Surveying and Mapping, Fall Tech. Mtg., Albuquerque, New Mexico, p. 56-69.

CIA, (1974), CAM: Cartographic Automatic Mapping, Program Documentation—Version 4, U.S. Central Intelligence Agency, Wash., D.C., 111 p.

Cederberg, R., (1980), "A Data Structure for a Raster Map Data Base," Proc. Intl. Symp. on Cartography and Computing (AUTO-CARTO IV, Reston, VA, p. 85-92.

Codd, E. F., (1970), "A Relational Model of Data for Large Shared Data Banks," Comm. ACM, Vol. 13, #6, June, p. 377-387.

Date, C. J., (1972), "Relational Data Base Systems: A Tutorial," from Information Systems COINS IV, 4th Computer and Info. Sciences Symp. Proceedings, Plenum Press, New York, p. 37-54.

Davis, J. C. and Lopez, S. L., eds., (1978), Computer Mapping for Resource Analysis, Proceedings of the 1978 COGEODATA Intl. Conf., Mexico City, Mexico, 307 p.

Deetz, C. H., and Adams, O. S., (1969), Elements of Map Projection, Greenwood Press, New York, 226 p.

Diello, J., (1972), "Experimental Cartography at the Rome Air Development Center," RADC, Rome, NY.

Dueker, K. J., and Horton, F. E., (1971), "Urban Change Detection Systems: Status and Prospects," Institute of Urban and Regional Research, Univ. of Iowa.

Dutton, G. H., ed., (1977), "Navigating Odyssey," in Proc. Advanced Study Symp. Topological Data Structures for Geographical Information Syst.," Harvard Univ. Laboratory for Comp. Graphics and Spatial Analysis.

Haralick, R., and Shapiro, L., (1980), "A Data Structure for a Spatial Information System," Proc. Intl. Symp. Cartography and Computing (AUTO-CARTO IV), p. 291-303.

Kellaway, G. P., (1946), Map Projection, Butler & Tanner Ltd., London, 127 p.

Liebenberg, E., (1976), "SYMAP: Its Uses and Abuses," Cartographic Journal, Vol. 13, #1, p. 26-36.

Maling, D. H., (1973), Coordinate Systems and Map Projections, George Philip & Son Ltd., London, 255 p.

Marble, D. F., Brassel, K., Calkins, H. W., and Peuquet, D. J., (1978), Computer Software for Spatial Data Handling--1978, for the IGU Comm. on Geog. Data Sensing and Processing, Ottawa, Ontario, Canada.

Mitchell, W., Guptill, S., Anderson, K. E., Fegeas, R., and Hallam, C., (1977), "GIRAS: A Geographical Information Retrieval and Analysis System for Handling Land Use and Land Cover Data," U.S.G.S. Professional Paper 1059, 16 p.

Mueller, C.S.M., and Christensen, A.H.J., (n.d.), "SCANLINE: A Computer Program for Converting Raster Maps into Line Maps," presented to Geoprocessing, T. K. Peucker, ed., 35 p.

Nagy, G., and Wagle, S., (1979a), "Geographic Data Processing," Computing Surveys, Vol. 11, #2, p. 139-181.

Nagy, G., and Wagle, S., (1979b), "Approximation of Polygonal Maps by Cellular Maps," Comm. ACM, Vol. 22, #9, p. 518-522.

Nordbeck, S., and Rystedt, B., (1972), Computer Cartography, Studentlitteratur, Lund, Sweden, 306 p.

Peucker, T. K., and Chrisman, N., (1975), "Cartographic Data Structures," The Am. Cartographer, Vol. 2, #1.

Peuquet, D., (1979), "Raster Processing: An Alternative Approach to Automated Cartographic Data Handling," American Cartographer, Vol. 6, #2, p. 129-139.

Rhind, D., (1974), "An Introduction to the Digitizing and Editing of Mapped Data," from Automated Cartography, British Cartographic Soc., London, p. 50-68.

Richardus, P., and Adler, R. K., (1972), Map Projections, North-Holland Publ. Co., Amsterdam, 174 p.

Roblin, H. S., (1969), Map Projections, Edward Arnold Ltd., Great Britain, 60 p.

Rosenfeld, A., (1977), "Extraction of Topological Information from Digital Images," presented at the Advanced Study Symp. on Topological Data Structures for Geographic Information Systems, Dedham, MA, Oct. 1977, Lecture Notes, 31 p.

Rosenfeld, A., and Samet, H., (1980), "Tree Structures for Region Representation, " AUTO-CARTO IV, Proc. of the International Symp. on Cartography and Computing, Applications in Health and Environment, Am. Cong. Surv. and Mapping and Am. Soc. Photogrammetry, p. 108-118.

Ross, C. G., (1980), "GIMMAP: A (Mini) Computer-Assisted Cartography System," Proc. of the Intl. Symp. on Cartography and Computing, (AUTO-CARTO IV), held in Reston, VA., Nov. 1979, p. 372-378.

Ross, C. G., and Brentano, J., (1980), "Groundwater Management District No. 4," a base map produced for the Geohydrology Section, Kansas Geological Survey, Lawrence, KS, 1:250,000.

Ross, C. G., McClain, T. J., and others, (1980), "Geologic Map, Lawrence West, Kansas," Map M-14, Kansas Geological Survey, Lawrence, KS, 1:24,000.

Steiner, D., (1979), "A Mini-Computer-Based Geographical Data Processing System," presented at the NATO Advanced Study Institute on Map Data Processing, Maratea, Italy, Lecture Notes, 15 p.

Struck, R. L., (1975), "Making a Map with Digital Data," Defense Mapping Agency Topographic Center, Washington, D.C.

Teicholz, E., and Dorfman, J., (1976), Computer Cartography World-Wide Technology and Markets, Intl. Technology Marketing, Newton, MA.

Tobler, W., (1962), "A Classification of Map Projections," from Ann. of the Assoc. of Am. Geog., Vol. 52, #2, p. 167-175.

Tomlinson, R. F., ed., (1972), Geographical Data Handling, IGU Comm. on Geograph. Data Sensing and Processing, 1327 p.

Tomlinson, R. F., Calkins, H. W., and Marble, D. F., (1976), Computer Handling of Geographical Data, UNESCO Press, 214 p.

Ullman, J. D., (1980), Principles of Database Systems, Computer Sciences Press, Potomac, MD, 379 p.

Zloof, M. M., (1977), "Query-by-Example: A Database Language," IBM Syst. J., Vol. 16, #4, p. 324-343.