

A COMPUTER SYSTEM FOR A
CADASTRAL MAPPING APPLICATION

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Master of Science in Computer Science
in the
University of Canterbury
by
M. S. COLLEY

University of Canterbury

1979

CONTENTS

CHAPTER	PAGE
ABSTRACT	1
1. INTRODUCTION	2
1.1 Purpose	2
1.2 General Structure	3
2. A MAP PROCESSING PROBLEM	6
2.1 Terminology	6
2.1.1 Automated mapping and Computer mapping	7
2.1.2 Cadastral maps	8
2.1.3 Thematic maps	8
2.1.4 Appellation systems	8
2.2 Elements of a mapping system	9
2.2.1 Procedural steps	9
2.2.2 Hardware types	11
2.3 New Zealand Department of Lands and Survey	13
2.3.1 Background	14
2.3.2 Computer mapping requirements	15
2.3.3 Hardware configuration	16
3. MAP PROCESSING SYSTEMS	19
3.1 Introduction	19
3.2 British Ordnance Survey	19
3.3 The Canada Geographic Information System	22
3.4 CLOTS	23
3.5 GIMMS	25
3.6 Polygon Information Overlay System	27
3.7 Summary	28

4.	DATA BASE DESIGN	30
4.1	Requirements for a Data Base	30
4.2	Cartographic Data Structures	31
4.2.1	Cartographic Spaghetti	32
4.2.2	Location Lists	32
4.2.3	Point Dictionaries	34
4.2.4	Polygon system	34
4.2.5	DIME	34
4.2.6	GEOGRAF	35
4.3	Proposed Structure	35
4.3.1	Data Base items	37
4.3.2	Specific Data Base requirements	37
4.3.3	Detailed design	38
5.	AN INTERACTIVE SYSTEM	47
5.1	Interactive Digitising	48
5.1.1	Advantages	48
5.1.2	Menu Driven Digitising	49
5.1.3	Map Orientation	54
5.1.4	Windowing	55
5.1.5	MULTIMAP	57
5.1.6	Digitising sequence	62
5.1.7	General Digitising Process	63
5.2	Editing and Updating	66
5.2.1	Editing	67
5.2.2	Updating	67
5.2.3	Interactive Language	68
5.3	Plotting Subsystem	73
5.3.1	Introduction	73
5.3.2	Arc Generation	74
5.3.3	Chain Encoding/Decoding	77
5.3.4	Character Set	80

5.3.5	Visual Centres	86
6.	INFORMATION PROCESSING SYSTEM	91
6.1	Small Scale Map Production	91
6.2	Additional Data Sets	95
6.3	Data Manipulation Subsystem	97
6.3.1	Feature Display	97
6.3.2	Feature Aggregation	97
6.3.3	Query Language	99
7.	CONCLUSION	101
	ACKNOWLEDGEMENTS	105
	REFERENCES	106
	APPENDICES	108

LIST OF FIGURES

FIGURE		PAGE
2-1	Configuration of Computer Hardware	18
4-1a	Example of a Location List	33
4-1b	Example of a Point Dictionary	33
4-2	The GEOGRAF Data Base Structure	36
4-3	Structure of the CHAIN DICTIONARY	41
4-4	Format of a POINT file word	42
4-5	Overview of the MAPPAK Data Base	45
4-6	Flow chart for interpreting XY DICTIONARY codes	46
5-1	Digitiser setup for Multiple Base Maps	59
5-2	Example of generating the 4th co-ordinate for the affine transformation matrix	60
5-3	Example of a digitising sequence	62
5-4	Interactive VDU display of an arc generation . .	65
5-5	Format of Interactive Digitising Screen displaying possible errors	65
5-6	XY DICTIONARY codes before and after deletion of an arc	71
5-7	Example of non coincidence of common points . .	72
5-8a	Example of reproducing chain encoded data . . .	78
5-8b	Example of effects of smoothing algorithm applied to chain encoded data	79
5-9a	Interactive VDU character set	84
5-9b	Flatbed Plotter character set	85
5-10	Example used for calculating Visual Centres . .	88
5-11	Transformation matrices	90
6-1	Small scale map production	92
6-2a	Drawing logic for an arc that starts and ends on the same sheet	94
6-2b	Drawing logic for an arc that starts and ends on a different sheet	94
6-2c	Drawing logic for connecting chain features across sheets	95

ABSTRACT

In 1973, the New Zealand Government instituted a metric measurement system and the New Zealand Department of Lands and Survey (NZLS) implemented a new mapping grid and projection system that permits the accurate representation of the whole country on a uniform basis. These two related decisions required the mapping division of the NZLS to assume responsibility for the conversion of both topographical and cadastral maps in the former imperial mapping grid system to the new metric mapping grid system.

This thesis deals with the design of an automated mapping system and identifies the techniques necessary to automate the conversion of cadastral maps from the imperial mapping system to the new metric system. The research undertaken encompasses the following three major areas:

- (a) the design of an efficient system to automate the conversion from one mapping series to another,
- (b) the identification and testing of techniques necessary to generate computer produced maps to cartographic standards,
- (c) to design a data structure for graphical data and its corresponding descriptive data in a format suitable for use by a land information processing system.

Although this research deals with a specific problem, the problems discussed in this thesis are of relevance to any general mapping application.

CHAPTER ONE

INTRODUCTION

1.1 PURPOSE

This thesis is concerned primarily with the application of computers in cadastral mapping. Cadastral maps are used to portray survey plan data approved by the New Zealand Department of Lands and Survey. Plan data comprises both a graphical description of the land parcels associated with each plan, and the corresponding legal attributes such as plan and section numbers. The research was undertaken in conjunction with the requirements of the New Zealand Department of Lands and Survey. A software system, referred to as MAPPAK (MAPing PAckage), was developed to demonstrate the ideas presented in this thesis. MAPPAK is now at the stage where it is being used efficiently for cadastral map compilation.

This thesis:

- (a) identifies the procedures necessary in the automation of the mapping process,
- (b) explains the purpose of those procedures,
- (c) describes the techniques used in the implementation of a computer-assisted cadastral mapping system.

The work of this thesis centres around the conversion from an existing manual cadastral mapping system to that of a computer-assisted system in which it is possible to produce both cadastral maps and thematic maps with a computer. Cadastral maps must be of a cartographic standard whereas thematic maps have less stringent accuracy requirements but greater information content.

The essential elements of an automated system are:

- (a) data capture subsystem,
- (b) editing subsystem,
- (c) update subsystem,
- (d) plotting subsystem.

While the system has a cadastral background, much of what is discussed here applies universally to any mapping situation. Common problems are identified and solutions to these problems are proposed.

The design of MAPPAK was influenced by two major factors:

- (a) the implementation of the experimental system was restricted to the hardware system described in section 2.3.3.
- (b) the design was to result in a system that could be used efficiently on a mini computer by inexperienced persons with minimal training.

Therefore, the work here has been influenced by factors that relate to the efficient use of a set of limited resources. This does not, however, preclude the use of the results in larger systems.

1.2 GENERAL STRUCTURE

Before designing a system, an understanding of the fundamental components of the system is required to define the system's "boundaries". Chapter Two introduces a mapping system by establishing the difference between Computer Mapping and Automated Mapping. The components of the system are then identified as a set of sequential processes and their functions, followed by an introduction to the particular pieces of hardware required for a computer mapping application. Finally, reference to the New Zealand Department

of Lands and Survey (NZLS), for whom this research was undertaken, serves to define the scope of the system.

Chapter Three then reviews some existing mapping systems, few of which are cadastrally oriented, and critically analyses the techniques used in terms of cadastral mapping requirements and the requirements of the NZLS. The discussion here applies equally to any mapping application and identifies possible approaches to the design of any computer mapping system.

A common feature in any mapping application is the data base. The data base contains files that comprise both the graphical information of a map and its corresponding descriptive data sets. For cadastral maps, the descriptive data files serve merely as a means of automatically including the text on a computer produced map. In thematic mapping, these files are organised to permit detailed analysis of geographic data and subsequent display of selected information. However, the author contends that the two file types need not be mutually exclusive, provided that the correct data structures are used. Chapter Four looks at the general types of structure available, and then at specific cartographic structures used in other systems. Finally, a detailed description of the author's structure is presented.

Chapter Five then describes the processes associated with data manipulation: how data is entered into the data base, edited for any inconsistencies, updated to reflect changes in the data base, and finally output to graphical display devices. Graphical data is obtained by an interactive digitising procedure that allows the user to monitor constantly his digitising progress on a display terminal. Error detection and correction functions are performed during

the digitising process. Some intermediate editing may be necessary before plotting the map. The types of error which may not be detected during the interactive digitising phase are presented, with a description of the interactive editing procedures used to amend the incorrect graphical data files. No detailed discussion on editing descriptive data files is necessary as this function is a clerical procedure requiring little technical background. Much of the discussion on editing relates equally to the update requirements of the system. For example, the edit procedures necessary to include a missing parcel are also used by the update process when inserting new parcels.

Finally, Chapter Five discusses the production of cadastral maps that are both aesthetically pleasing, and of a standard comparable with a hand drafted cartographic document. It describes the techniques and algorithms used to produce smooth arcs and cartographically acceptable chain reproduction. Techniques for presenting non positional data are discussed. Both its physical location and appearance is a major factor contributing to the acceptability of a computer-produced cartographic document.

Chapter Six then examines the means by which the cartographic data can be added to and manipulated, to produce special purpose maps and display geographical relationships. It discusses techniques to produce small scale maps from the data base's large scale base maps, as well as techniques for manipulating cartographic and geographical data to produce thematic maps.

The final chapter summarises the conclusions derived from the experimental system MAPPAK, which is now currently being used for cadastral map compilation.

CHAPTER TWO

A MAP PROCESSING PROBLEM

Computer-assisted mapping systems are needed for three basic reasons:

- (a) to satisfy the ever increasing demands for land-related information,
- (b) to provide an economic method for rapid processing of the large amounts of input to manual land information systems,
- (c) to assist in the conversion to, or adoption of, a cadastral mapping base.

In every country, the need is now recognised for a system that is capable of producing land-related information to satisfy requests from both individuals and organisations. Countries with a formal cadastral system have already begun to undertake some computer-assisted mapping applications. Those lacking a cadastral base for mapping purposes have first to consider the transition from their present ad-hoc system to a cadastral base. Irrespective of the presence, or lack, of a cadastral base, a land system capable of fulfilling the above requirements implies a need for a multipurpose cadastre. Such a system should be capable of producing up-to-date maps, and should correlate data on a parcel basis with current parcel information such as legal status, ownership, valuation, and usage.

2.1 TERMINOLOGY

Definitions are now given for some of the terminology used by the author.

2.1.1 Automated mapping and Computer mapping

Taylor (1973) found it necessary to distinguish between automated mapping and computer mapping. He defined automated mapping as the automation of the map making process - the implication being that maps produced from automated mapping were not inherently different in style or presentation to existing maps. His definition of computer mapping referred to the production of maps using the analytical power of the computer to display spatially oriented relationships. Often, the outputs of computer mapping were thematic maps whose presentation neither reflected nor was required to reflect the production standards of automated maps.

While this distinction may have been applicable in 1973, the author believes that it is no longer necessary because

- (a) the application of completely automated scanner digitisers (Section 2.2) remove the need for manually digitising source documents, and
- (b) a cartographic data base can form the basis of a system.

The cartographic data base can provide the elements for producing maps of a cartographic standard as well as for deriving thematic maps. In Chapter Six, the author describes in more detail how a cartographic data base can be extended for the purposes of thematic mapping.

Therefore, by "automated mapping", we refer here to a process of map production which is completely automated. We shall use "computer mapping" to refer to a system requiring manual intervention but capable of producing both cartographic and thematic maps.

2.1.2 Cadastral maps

In its simplest form, a cadastral map presents a positive identification of all land parcels and their legal status. It is compiled from survey plans that comply with strict accuracy standards. Increasing demands for better land use and greater planning to cater for expanding population requirements, have led to the use of cadastral maps for such purposes as the planning and administration of housing, commercial and industrial development, and the protection of the environment. Consequently, the combining of fiscal cadastre and juridical cadastre could create a multipurpose cadastre capable of producing maps of cartographic quality as well as containing the elements essential for a planning system. The success of such a system is dependent on the accuracy of the cadastral base map, its translation to digital format, and its reproduction by a computer.

2.1.3 Thematic maps

A thematic map portrays primarily spatial data in the form of relationships between different logical entities. For example, logical entities may define soil, vegetation or rainfall distributions, population densities, or any other similar type of entity. Strict cartographic standards do not apply for thematic maps, due mainly to the difficulties in accurately defining the boundaries of thematic entities.

2.1.4 Appellation systems

An appellation system attaches to every parcel a legal description and a means by which the parcel can be located on the appropriate map. Within any land information system, it is essential that all parcels have a unique identifier (Moyer 1973) so that the legal description of the land permits:

- (a) identification of the approximate geographical location,
- (b) identification of the parcel in such a way that it cannot be mistaken for another,
- (c) display of the dimensions of the parcel.

In a computer system, a unique appellation or geocoding system is needed as a source of secondary information by which individual parcels can be referenced. In New Zealand, although several appellation systems exist, all parcels have a unique appellation. However, a common appellation identifier is essential in a computer system to permit the many aspects of land information to be successfully related to the parcel.

2.2 ELEMENTS OF A MAPPING SYSTEM

All computer map processing systems have several functions in common. Their fundamental elements can be summarised as: data acquisition, storage, manipulation and display. Depending on what type of hardware configuration is used, several methods exist for performing each function. The decision on how best to accomplish the basic processes depends primarily on the nature of the system and its general requirements, and not on the types of hardware available.

2.2.1 Procedural steps

The functions of data acquisition, manipulation and plotting represent the flow of a document through a mapping application. In the data acquisition phase, there are at least three methods available, all of which differ in both their resource requirements and efficiency.

- (a) Off-line Manual Digitising (Monmonier 1970) is

widely used to produce co-ordinate data. After digitising, a lot of post-processing is required to correct the many errors that are possible, for example, missing segments, faulty references, and non-coincidence of end points. As a result, this process is costly and inefficient.

(b) Online Interactive Digitising (Kroll 1974) is at present the most economical way, provided that the quality of the output on the interactive display device is reasonably close to the standard required for the final product. Online systems ensure that lengthy post-processing sessions to remove errors are avoided.

(c) Automated Scanning (Boyle 1978) produces large amounts of geometric data requiring complicated processing of the raster data to generate the data base files. This process is both difficult and expensive.

The storage function involves adding input data to the data base. This function should be transparent to the user as it is concerned only with the internal processing of the external data. In an interactive system, this function is performed in real-time. On satisfactory completion of data acquisition, the data base should contain the new digital data without the need for post-processing to organise the data base.

The manipulation phase processes stored data which involves:

- (a) editing the data base to remove any inconsistencies between the source data and its internal representation,
- (b) updating the internal data to reflect the daily

- changes in the external data it portrays,
- (c) processing the internal data structures to display the relationships between different geographical attributes.

Finally, the plotting phase processes the data files to produce documents requested by the users. This may result in either a special purpose map displaying some phenomena derived from a request as in (c) above, or merely a plot of the computer's copy of the source document.

2.2.2 Hardware Types

We now look at the two functions of input and output, and the types of hardware available to accomplish these functions.

A) Input. There are several methods for transcribing existing source documents to digital format:

- (a) Digitisers are now used for the bulk of cartographic input because of their low cost, ready availability, and ease of use. A digitiser establishes geometric locations in terms of x and y co-ordinates relative to the digitiser table axes. The digitised data is processed into a form suitable for inclusion in the cartographic data base.
- (b) Stereographic Plotters have now been modified to produce digital output, mainly to paper tape for input to a computer system.
- (c) Raster Scanners (Boyle 1978) use a wholly automated procedure for reading map information into a raster matrix and then converting it to vector format by software. Unfortunately, maps are composed of many different features of varying line widths. As a

consequence, the software needed to differentiate between features is very complex. To overcome this obstacle, maps are preprocessed to separate features, and each feature is independently scanned and later merged with the other features.

- (d) Automatic Line Followers (Turke 1978) use an optical array camera mounted on a flatbed plotter and interfaced to a minicomputer. The camera is positioned manually over the areas to be digitised and it then automatically follows the surrounding segments. This method suffers from several technical difficulties. For example, whenever an intersection point for two possible paths is detected, the system requires that the operator decide upon the correct path to take. In the case of processing annotated maps, the features must be separated from the text prior to processing.

B) Output devices are of four types:

- (a) Line Printers are often used for map display, but are limited in their display capabilities. Maps are restricted to having a character print position represent a grid cell, symbols are limited by the character set of the printer and to a fixed size. Area definitions are possible only through choices of characters and overprinting to obtain the correct densities. Approximation of curved lines is possible only through a connected chain of printer cells. Generally, poor reproduction of line features occurs. Printers are therefore not suitable for cartographic maps but may be used to produce thematic maps.

- (b) Plotters have the ability to draw continuous lines and curves, but are not really convenient for shading regions to display toning patterns. Although it is possible to differentiate areas by shading with different patterns, complicated procedures are required to calculate the boundary points of the areas to be shaded. Further differentiation of features is possible through the use of colour. Through appropriate software, different pens can be selected to provide either different colours or changes in linewidth.
- (c) Display Devices can produce maps faster than either printers or plotters. They generally have relatively small screens. Interaction with the screen is via a cross hair cursor, for example, which causes the xy co-ordinate pair of the identified screen location to be generated. There are two types of display devices - storage tubes and refresh tubes. On a storage tube, the information is written once only and the image is retained on the surface. The refresh screen, if a raster scan, has a cathode ray beam that repeatedly scans the entire screen line by line. The storage scope gives good resolution and is flicker free, but it is not possible to selectively erase parts of the screen as is the case with the refresh screen.
- (d) Laser Beam Displays are a recent addition to display devices. The LASERSCAN HRD-1 (Street 1975) is one such device that is controlled by a mini computer and information is plotted on a frame of

photochromic film cartridge which is continually projected onto a large screen.

2.3 NEW ZEALAND DEPARTMENT OF LANDS AND SURVEY

The research work described in this thesis was done for the New Zealand Department of Lands and Survey. This section introduces their mapping requirements and describes the hardware used for the experimental system.

2.3.1 Background

The cadastral map has the primary function of displaying land parcel boundaries annotated with their appellation, area and survey plan reference. Its origins date back more than 100 years. During this time, land surveys have been integrated in a single survey system and administered by a single body - the NZLS. With the land parcel as the basic unit, land title surveys are co-ordinated on a plane grid, the survey plans prepared, and the cadastral maps produced to display accurately the boundaries and description of every land parcel. Besides the compilation of the cadastral map, the Department must also provide an updated record through the daily plotting of parcels and other information created by approved survey data.

2.3.2 Computer Mapping Requirements

With a well established cadastral base, the requirements for land administration and planning have resulted in a demand for greater cadastral coverage of the country on scales larger than previously recorded. Simultaneously, a new single grid using a metric projection system was adopted by the NZLS. The result has been a requirement to convert existing imperial cadastral maps

compiled from 12 different survey districts to the new metric mapping grid. Continual demands for specialised maps involving the conversion and adoption of existing maps for purposes such as police geocoding schemes, statistical and population analyses, and valuation recording, as well as the requirement to convert to the new mapping grid, have provided the impetus for a computerised mapping system in New Zealand. Such a system will eventually give the benefits of:

- (a) a large central data base,
- (b) the ability to manipulate and interrogate the data base,
- (c) immediate updating facilities,
- (d) the ability to provide special purpose maps at a variety of scales,
- (e) automatic conversion from an existing imperial format to the new metric base map.

Since data is to be initially captured and subsequently updated on a cadastral map basis, the map sheet will form the basic unit within the system. The new cadastral sheets are held at scales of 1:1000, 1:2000 and 1:10000 and are drawn within a 50cm x 75cm frame. The features on a sheet are land parcels, roads and railways, and hydrographic features. A feature is defined by its graphical form as well as its supplementary descriptive data. All survey points on New Zealand cadastral maps are restricted to:

- (a) straight lines between points, for example, a road boundary,
- (b) surveyed arcs lying on the perimeter of a circle, for example, the turning circle at the end of a no exit street,
- (c) completely arbitrary curves that represent some

natural feature, for example, a coastline.

Every land parcel has associated with it a plan number, lot or parcel number, an area, and a certificate of title (C/T) reference. In some instances, it is necessary also to attach to a parcel a gazette reference, which identifies the page number and year of the gazette that describes the land's use, for example, a recreational reserve or police station.

To incorporate descriptive text on the cadastral map, we need some form of logical association between graphical entities and their descriptive data. If this association can be provided for in a well designed data base, then it would be possible to use this cartographic system as the basis for a more generalised land information system. This would then provide the capacity to interrogate and display certain attributes in the data base and produce specialised maps such as police geocoding schemes and census evaluation tracts.

Also, to assist in the conversion from the imperial grid to the metric grid, the system must provide an effective method for recording data from several imperial source documents and generating a single metric base map.

2.3.3 Hardware Configuration

The NZLS hardware configuration for the experimental system is shown in figure 2-1. The graphical hardware components are:

- (a) DIGITISER: which is controlled by a micro processor and has inbuilt area, length, and skew functions. When the cross hair cursor or pen type stylus is used, two of the six digitising modes used are:

- i) POINT: Single co-ordinates are recorded on activation of the sampling switch,

ii) SWITCHED STREAM INCREMENTAL: Co-ordinate pairs are continuously updated if the last point recorded differs from the previous point by a user definable amount. The capture rate varies between 1 and 100 conversions per second depending on the user selected setting.

(b) TEKTRONIX: which is a storage screen equipped with a cross hair cursor and keyboard. It has 1023 addressable points on the X axis and 780 on the Y axis.

(c) PLOTTER: which is on-line to the PDP 11/34 and has a drawing surface of 78cm x 92cm. It is capable of plotting high quality straight lines at 56cm per second. Having a high repeatability factor, it is suitable for producing high quality cartographic documents.

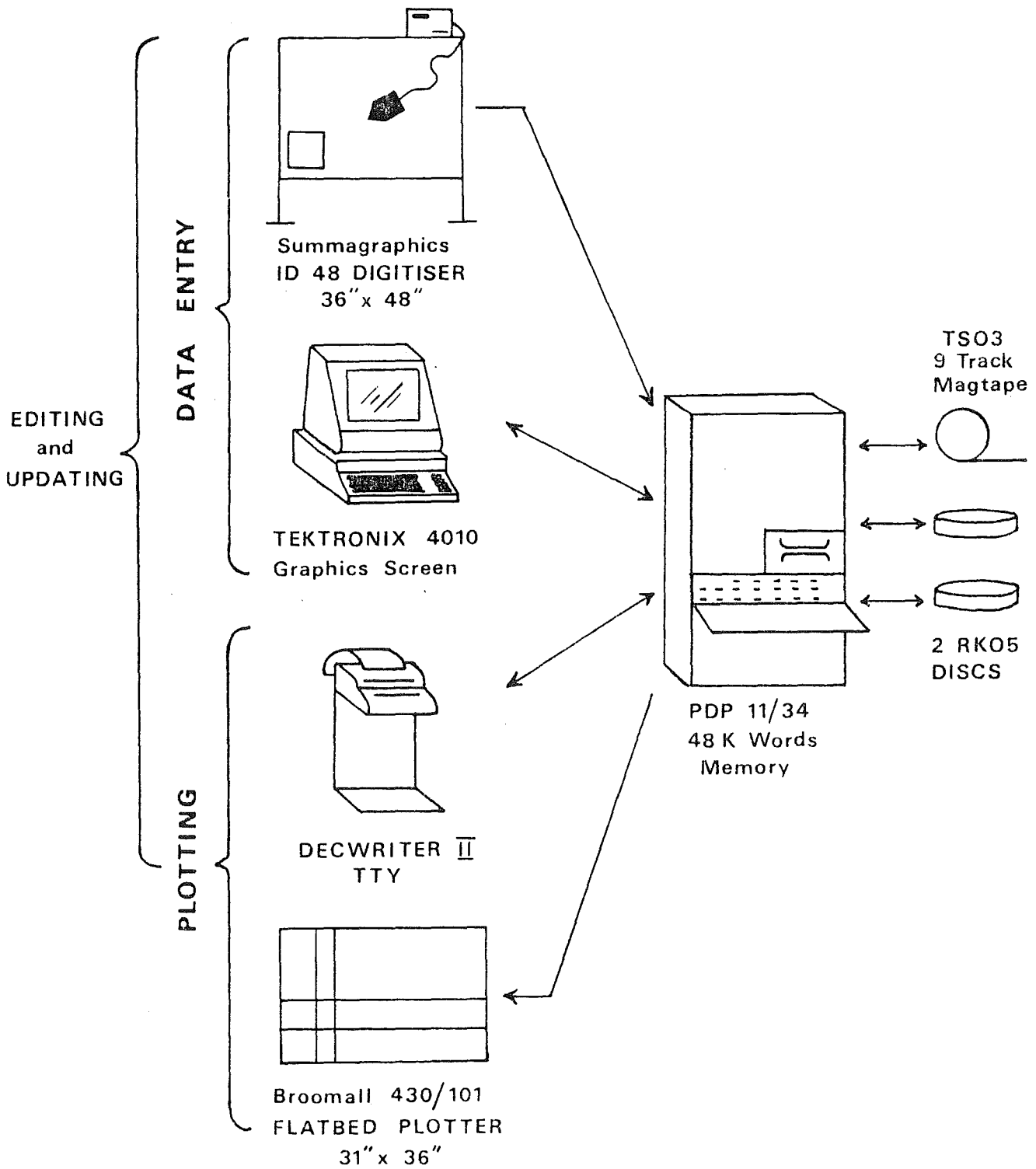


Figure 2-1
CONFIGURATION OF COMPUTER HARDWARE

CHAPTER THREE

MAP PROCESSING SYSTEMS

3.1 INTRODUCTION

This section reviews a number of mapping systems and their specific application areas. A lack of formal cadastral mapping bases in many countries results in a general absence of cadastral systems such as that described in this thesis. Therefore, the systems studied range from the very specific applications area to large general purpose geographical processing systems. The majority of systems developed are concerned primarily with information processing of both statistical and spatial data. None of the systems studied was designed to serve the dual purpose of providing both a cadastral mapping base with a parallel geographical information data base, as no description of such a system could be found by the author.

The review of the systems is for the purpose of formulating a set of design principles, rather than for a critical analysis of the individual systems.

3.2 BRITISH ORDNANCE SURVEY

The prime function of the British Ordnance Survey (OS) (Aktey 1975) is the maintenance of nearly a quarter of a million large scale maps. In 1972, the OS embarked upon a project to study automated techniques for digitising large scale maps and to output the results graphically to the accuracy attainable by conventional manual means. Their system is geared to the production of 1:1250 and 1:2500 scale

maps.

The data capture subsystem uses manually controlled cartographic digitisers to convert graphical data to digital form, and stores it directly on magnetic tape. Source documents are photographically enlarged by a 3:5 ratio to provide maximum utilisation of the digitising surface. All input is discrete points. No attempt is made to stream digitise by following lines while periodically recording co-ordinates. Points along curved lines are digitised at the operator's discretion and a mathematical spline is applied in subsequent processing to generate the additional points needed to give a continuous curve. The generated points are then stored in the data base. The following reasons were given for adopting this approach rather than stream digitising:

- (a) following lines with a cursor is difficult, and therefore inaccurate. The gain in accuracy is achieved through skilful selection of points for the spline algorithm,
- (b) it is less tiring for the operator,
- (c) selection of points puts skill, and therefore satisfaction, into digitising,
- (d) less complex digitisers are required,
- (e) it is more economic to store only as much data as is absolutely necessary to represent a line.

However, the author will describe a technique disputing the assumption made in (a), and describe alternative techniques to reduce the storage requirements for stream co-ordinates.

Before digitising, the document is pre-edited to assign sequential numbers to all name and number positions on the document. The digitising sequence requires these locations to be digitised in the order of their numbers with the text being

added at a later stage on specially designed forms.

All features are categorised by a descriptor denoting their feature code. Prior to digitising a feature, the feature code is recorded by digitising a point in a pre-assigned menu area on the digitiser. The operator must manually keep a record of digitised features to prevent redigitisation. On map completion, the magnetic tape data is processed and added to the OS data base. Data is held on a map sheet basis with a header record containing information about scale and origin. Each sheet is further subdivided into 1024 parts in both the x and y directions to:

- (a) allow areas of a conventional size to be extracted and plotted for specialised maps,
- (b) allow packing of xy values into one computer word.

The data processing involves such items as:

- (a) converting digitising table co-ordinates to tranverse mercator co-ordinates,
- (b) producing points along a spline,
- (c) generating a parallel line for a railway.

An editing operation is necessary to check data for "positional accuracy, completeness and correct feature coding" (Atkey 1975). Accuracy and completeness are checked by superimposing the graphical plot over the original document. A printer document lists feature serial numbers and their descriptions to be matched with the plotter document.

Missing features are digitised while incorrect features are edited by submitting their serial number and edit commands on a card deck. The editing commands are:

- (a) CODE: replaces a feature's code,
- (b) DELETE: removes a feature from the data base,
- (c) JOIN: extends or truncates a line to reconcile its

- junction with the boundary of another feature,
- (d) PJOIN: replaces the end co-ordinate pair of one feature with the co-ordinates of a second feature, such that the two co-ordinates are the same.

Updating is implemented through the same mechanism.

The plotting subsystem uses a plotter fitted with a light spot projector which plots directly onto photographic film. Variable line widths are caused by changing the projector's aperture during the plot. After a few manual operations which include the addition of slope and ornamentation symbols, a printing plate is generated as the final output from the system.

3.3 THE CANADA GEOGRAPHIC INFORMATION SYSTEM

The Canada Geographic Information System (CGIS) was designed to "facilitate the use of data collected by the Canada Land Inventory system" (Tomlinson 1976). Its primary concern is with mapping the availability of land for various uses. Although the system has a graphics display capability, the emphasis is on the production of tabular displays.

The main storage unit is called a 'coverage' which comprises data on such items as:

- (a) soil suitability for agriculture,
- (b) land capability for forestry,
- (c) census boundaries,

as well as other land related information. Coverage data is held on a polygon basis in the system and its descriptive data is entered manually.

Image data in the form of polygons is entered by one of two processes:

- (a) automatically by a drum scanner, or by

(b) offline digitising.

The scanner records one bit of information on a tape for every map square. A 76cm x 76cm map has 50 million map squares. Wherever lines are detected, a 1-bit is held for that square. A processing phase accepts 'clouds' of points and reduces them to lines one point wide using an algorithm called 'V-values' (Tomlinson 1973). The output from this process is basically a set of chain codes (see Chapter Four) representing the polygon. These are further processed and held in compact form in the Image Data Set file. Image data for each coverage is stored in an ordered hierarchy with descriptors linking all polygons of a specific coverage.

The retrieval and analysis of data requires the assistance of a programmer, a set of library programs, and any special purpose programs necessary to satisfy the request. The system will both plot maps and print tables produced by the manipulation subsystem. In 1975, the project designers were faced with the problem of automatically labelling maps without printing the labels on top of one another. The suggested solution was to move the label locations using an interactive display terminal. Another problem concerns detecting and discarding line segments which are not part of the polygons, but occur as a result of extraneous marks or dirt detected by the scanner. Finally, as a result of the method for holding data in the Image Data Set file, a saw-tooth effect is produced on the output of polygon boundaries.

3.4 CLOTS

CLOTS (CLOTS 1978) was developed to be used in conjunction with the Land Ownership and Tenure System (LOTS)

in Southern Australia. The objective was to develop a general acquisition system for land title office plans which would:

- (a) allow mathematical checking of lodged plans,
- (b) create a plottable file,
- (c) create a data bank of all the plans,
- (d) allow the amalgamation of plans or parts thereof to generate a plottable file.

The basic unit within the system, a lot, is any area on a plan for which the boundary is known. Such areas include roads, reserves and easements, all of which are restricted to straight line segments.

The input subsystem accepts surveyed lot data on a plan basis. Lot data must be entered clockwise around the plan. Data is either corner co-ordinates and their sequence numbers, or sequence numbers that delineate each lot, distances between them, and the internal angles within the lot. In the latter case, the data is processed to generate co-ordinates and sequence numbers which are held in a co-ordinate file and corner number file respectively. Both input and edit data is entered using an interactive command language that issues prompts for the required input. After an initial dialogue in which the user supplies file names, plan numbers and other initialisation data, the system is ready to accept lot data. The general format for input is

pt1/pt2:distance;bearing

where pt1 and pt2 are two points connected by a line. Variations of this format cater for instances where points are already defined. A set of 12 commands provides the means for building lot data records and editing existing records. No graphical display is available as the data is entered.

The output subsystem plots the plans. A number of

options including plotting point numbers, specified lots, lot numbers, and areas cater for various user requirements.

3.5 GEOGRAPHICAL INFORMATION - MANIPULATION AND MAPPING SYSTEM

The GEOGRAPHICAL INFORMATION - MANIPULATION and MAPPING SYSTEM (GIMMS) was designed by Thomas Waugh (1977) to be a general purpose geographic processing system, although its primary use is in thematic mapping. It incorporates the basic facilities of input, storage, and manipulation. A command language provides the user interface to the system which can run in batch or interactive mode. When using the system, the user generally goes through three phases:

- (a) establish locational data banks,
- (b) establish non-locational data banks,
- (c) manipulate the data banks to produce tables, diagrams and maps.

The locational data bank holds positional data related to geographical entities. It uses three types of geographical descriptions: points, line segments/networks, and area zones. The generation of descriptions is on a hierarchical basis. Locational data is prepared using a digitiser. Lines in a network comprise a segment label and the points identifying them. Area zones are constructed from line segments. Segment referenced networks must be created with labels for each line segment, reflecting the two areas for which it is a boundary. This mechanism for establishing locational data banks requires:

- (a) the digitising of co-ordinates in an anticlockwise direction around the zone corresponding to the left hand member of the pair of labels,

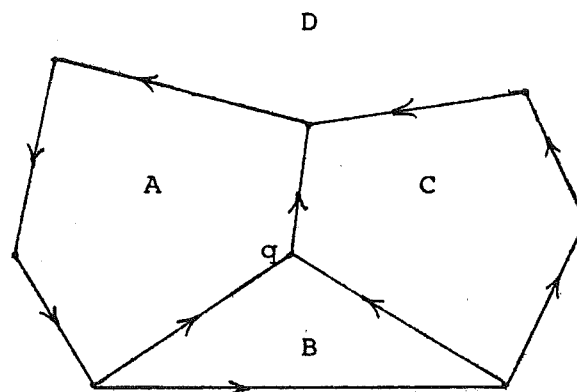
(b) complex algorithms to construct area files after having checked area zones to ensure geographical "completeness, correctness and consistency".

Example:

```

* TYPE=POINTS
  .
  .
  x y co-ordinate
  .
  .
* END
* TYPE=LINES
  /label1/ .....Xm Ym
  /label2/ .....Xn Yn
  .
  .
* END
* TYPE=AREA
  /A/B/ .....Xm Ym
  /A/C/ ....
  /A/D/ ....
  /B/C/ ..
  /B/D/ ....
  /C/D/ ..
* END

```



Point q, for example, is at the junction of three segments and will be digitised three times. The system requires that the co-ordinates of the point be equal for boundary checking and area file creation.

Non locational data banks contain numeric and alphanumeric information related to geographical zones. Up to 100 variables can be specified for a zone. User-chosen names are associated with elements in the data set and using arithmetic and logical operations, new data items are created from existing sets. For example, if

V1 = number of rented dwellings, and

V2 = number of owned dwellings

are two variables for a zone, then a new variable representing the percentage of total dwellings that are rented would be specified by

SET PCRENT (V1*100/(V1+V2))

Manipulation facilities also perform arithmetic and logical operations on a zone's non locational data set. Output from the manipulation system includes printer listings, tables, and histograms, printer maps using shading techniques, and graphical plots. A variety of commands are included to indicate the nature and format of the output, including interfaces to such plotting packages as GINO, CALCOMP and GERBER. The system includes a range of symbols and a character set with both super-script and sub-script capabilities.

The system in general suffers from the following problems:

- (a) symbol location uses a geographical centroid calculation which may generate a point outside a concave shaped zone,
- (b) appearance of plotted circles is a problem as the beginning and end points are visible, and small circles tend to be square,
- (c) non-locational data cannot be attached to line segments, but only to line networks,
- (d) hidden line removal is not implemented for overlapping regions.

Nevertheless, this system is used by a number of organisations throughout the world.

3.6 POLYGON INFORMATION OVERLAY SYSTEM

The Polygon Information Overlay System (PIOS), (Tomlinson 1977), was designed to provide data for computer models to serve local and regional planners in Canada. Data is stored as discrete polygons with each polygon described by

the co-ordinates of its vertices. Each polygon has classification data associated with it such as floodplain data, traffic zone data, or noise contour data for an airport. The system uses polygon overlays to calculate areas and produce primarily tabular results.

The land use inventory consists of maps showing polygons interpreted from aerial photography rather than land use by parcels. Data is entered via a digitiser and keyboard. The digitiser operator has to align each map so that the map's co-ordinate axes are parallel to the digitiser axes. This operation takes from 15 to 30 minutes for each map. Digitising is also hindered in that points can be digitised faster than the machine can read them and the digitiser origin often shifts due to equipment malfunctions, voltage variations, and insufficient insulation in the digitisers.

Once digitised, error correction requires redigitising erroneous polygons from proof plots used to highlight errors. Error free files are then merged for each classification data variable and ordered by map number, classification data value, polygon number and polygon type code. The type code identifies polygons completely contained within other polygons, which is required for computing the outer polygon's area.

Data retrieval and analysis comprises the measurement of polygon areas, and the overlaying of two sets of polygons with calculations of the intersection area.

3.7 SUMMARY

The aim in this section has been to discover the essential elements a mapping system should include, avoiding those constraints imposed by the systems studied that gave

rise to possible problem areas. This permits the compilation of a list identifying the tools necessary to implement a successful system. From the systems studied, the author derived the following conclusions:

- (a) Interactive systems provide better facilities for the user than off-line systems,
- (b) Dual cartographic/geographical processing systems are possible provided the cartographic data is in a form accessible to an information processing subsystem,
- (c) Preprocessing of input data prior to digitising, digitising sequence restrictions, and entry of descriptive data during digitising, should all be kept to a minimum,
- (d) Alignment of map axes with digitiser axes must not be necessary,
- (e) It can be expensive to store map co-ordinates as actual grid co-ordinates in preference to holding them as offsets from a false origin,
- (f) Co-ordinate compaction for stream oriented data is necessary to reduce file sizes,
- (g) Digitising from large scale maps is preferable to generating large scale maps from small scale data,
- (h) Some form of windowing is necessary for successful graphical manipulations,
- (j) Location of descriptive text requires user controllable centroids and/or visual centres in preference to geographical centroids.

CHAPTER FOUR

DATA BASE DESIGN

4.1 REQUIREMENTS FOR A DATA BASE

During the design of a data base, conflicting objectives must often be resolved. These conflicts are a result of:

- (a) the limitations of the available hardware,
- (b) the multiple requirements that the system must serve.

The data base design must result in a structure that organises the data elements in the files such that the logical relationships between data elements is maintained.

The following factors must be taken into consideration during the design phase, and are partly responsible for the conflicting objectives. No attempt has been made to organise the requirements in order of priority.

- (a) Retrieval. A general design principle is that all data be structured so that it can be retrieved "totally and flexibly" (Williams 1971).
- (b) Expansion. It is not possible to ascertain in advance all the requirements for the system; as the external environment changes so do the demands made on the system. If the structure does not provide sufficient flexibility it will not meet the changing requirements. The structure must be capable of expansion without the need for redesign. Expansion thus implies the ability to incorporate new data sets without restructuring the former system. We must include the new, as well as

preserve the present, logical associations between data elements.

(c) Access. The structure should provide for quick access to, and efficient updating of, data items. In general, a data base must allow for interrogation by data element attributes, for the selective searching of data elements and for the display of the results within a time period acceptable to the user. The structure must also allow for use of an efficient method for initial data entry and subsequent updating.

(d) Storage Requirements. Two types of structures:

- i) external files which hold the data on mass storage, and
 - ii) internal files which represent the memory resident copies of external files,
- must be considered in terms of storage requirements. At this stage the designer must make a compromise. Generally, storage requirements may be reduced at the expense of additional processor time to reconstruct the data element relationships.

The designer's problem is to specify a data structure which, for a given situation, is satisfactory in terms of the above factors.

4.2 CARTOGRAPHIC DATA STRUCTURES

Cartographic data bases contain two types of data:

- (a) co-ordinate data necessary for the production of linework,
- (b) text data describing cartographic features and their elements.

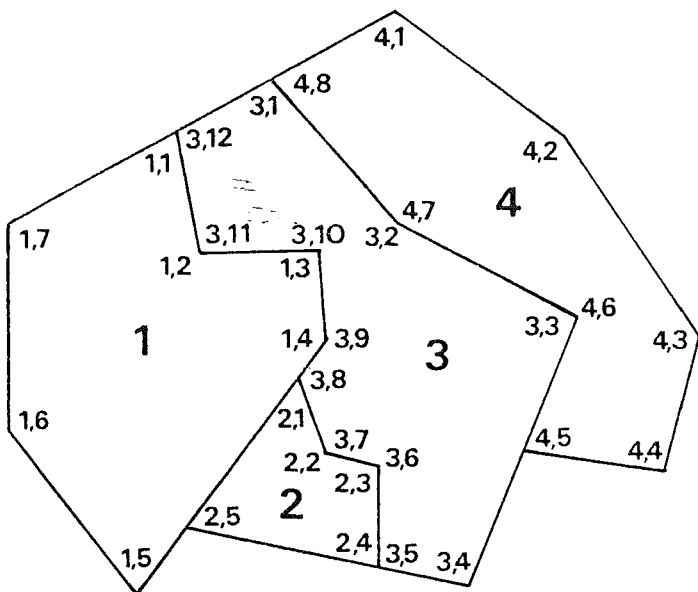
Here we review existing techniques for the organisation of co-ordinate data files and the methods by which cartographic features can be defined by their co-ordinates. The techniques range from the very simplest, referred to by Schmidt (1974) as cartographic spaghetti, to sophisticated, application-dependent structures such as DIME (Cooke 1967) and GEOGRAF (Peucker 1977). As well as the objectives described in 4.1, the evaluation of each type of structure should take into account the design criteria presented in section 4.3.2.

4.2.1 Cartographic Spaghetti

This refers to any elementary technique that generates an unstructured file of line data. The data cannot therefore be manipulated and is useful only for reproducing maps. Such a system was used by WORLD DATA BANKS I and II (Schmidt 1969).

4.2.2 Location Lists

With this technique, a feature is defined by specifying each of the co-ordinates around its boundary. No allowance is made for adjacent entities sharing common lines (see Figure 4-1). Such a system has several shortcomings. Boundary segments between adjacent entities have to be digitised at least twice, resulting in all but the boundary points being stored twice. The problem of 'sliver' lines arises, as we get lines duplicated in slightly different positions. Also, it is uneconomical to encode entities independently because of the time required after digitising to edit out any inconsistencies. Map Model (Arms 1970) is a system that uses the above technique but searches for segment lines that are common to two entities and removes one. Segments that have no duplicate are flagged as potential errors.

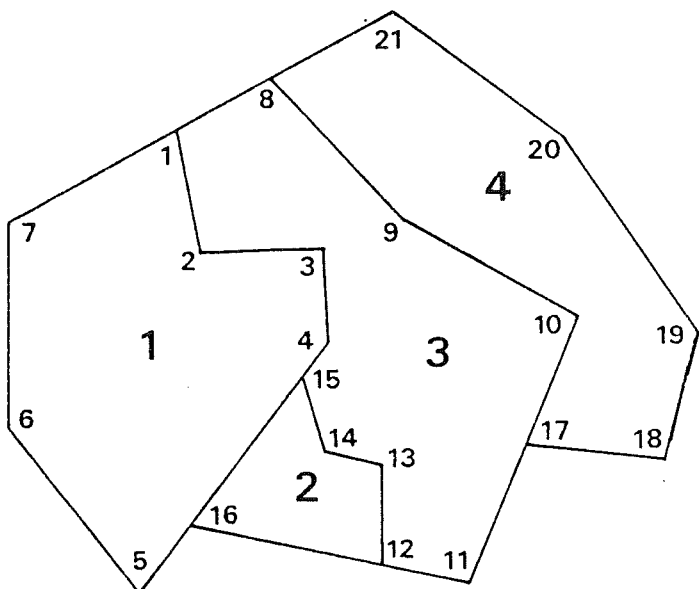


POLYGON 1 $X_{1,1}$ $Y_{1,1}$
 $X_{1,2}$ $Y_{1,2}$
 \vdots \vdots
 $X_{1,n}$ $Y_{1,n}$

POLYGON 2 $X_{2,1}$ $Y_{2,1}$
 $X_{2,2}$ $Y_{2,2}$
 \vdots \vdots
 $X_{2,n}$ $Y_{2,n}$

Figure 4-1a

Example of a Location List



POINTS X_1 Y_1
 X_2 Y_2
 \vdots \vdots
 X_n Y_n

POLYGON 1 1,2,3,4,5,6,7,1
 POLYGON 2 12,13,14,15,16,1
 POLYGON 3 1,2,3,4,15,14,13

Figure 4-1b

Example of a Point Dictionary

4.2.3 Point Dictionaries

This technique uses two structures: Point Dictionaries and Feature Boundary Lists. The Point Dictionary contains all the co-ordinates on the map, and a Feature Boundary List contains a list of all points that represent a given entity. This approach is used in INTERMAP (Peucker 1973) and GIMMS (Waugh 1977). The problem of 'sliver' lines is avoided but the plotting problem is compounded as lines are arbitrarily referenced. Unless some method for detecting duplicate lines is applied, this technique suffers from the duplicate plotting of lines.

4.2.4 Polygon System

This is a similar technique to the use of Point Dictionaries and it is used by the Polygon Information Overlay System (see Section 3.6). The method is to digitise neighbouring lines between two segments only once and store these as separate co-ordinates in a Segment Definition file. Polygons are then defined by a sequence of these segments indexed by a number. A negative number indicates that the segment was recorded previously. This method suffers from duplicate plotting of common boundaries.

4.2.5 DIME

The DIME file is an application-dependent structure that cannot be adequately used for cartographic systems. It was developed for the US Bureau of the Census to provide checking for errors in the address coding of the 1970 census. Streets are decomposed into straight line segments and each segment contains identifiers to:

- (a) the end co-ordinates of the segment,
- (b) the polygon features on either side of the segment,

(c) the high and low street addresses.

Files are very large and it is not possible to combine segments to form more complex objects.

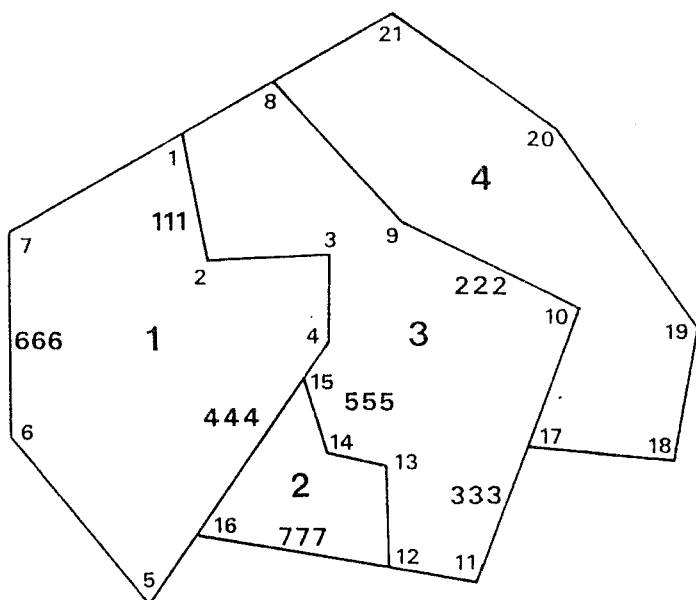
4.2.6 GEOGRAF

This system is similar to the DIME in that it defines a segment by a set of chains. Each chain has two nodes at its ends, separates two areas, but may be constructed from many points. However, for multiple polygon types that overlap, a chain that differentiates only two areas cannot remain the controlling unit. An additional structure called the Least Common Geographic Unit (LCGU) defines areas that are uncut by further partitioning and uses chains to define these areas. Polygons then comprise 1 to n elements from this new set, and a complex hierarchy of relationships can be built (see Figure 4-2). The data structure in section 4.3.3 uses a similar structure but uses a feature aggregation procedure (see section 6.3.2) to generate complex data sets.

Each of the methods described above has its particular disadvantages. In addition, none allows for graphical data types other than straight line segments. A requirement for a cartographic data base is that it can hold all needed graphical data types.

4.3 PROPOSED STRUCTURE

We now describe the structures developed by the author for generating a cartographic data base. The data elements are held as a combination of sequential structures, indexed structures, and linked lists. With these basic structures, a complex data base can be created in the form of a hierarchical, network, or relational type data structure. The



CHAIN GROUP

111
444
666
555

POLYGON

Pointer	Name
—	1
—	2

Name XY

1	
17	
15	

LCGU

111
222
333
444
555

Points from to left right

	1	15	3	1
	1	17	3	4
	17	12	0	3
	15	16	1	2

POINTS

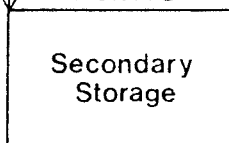


Figure 4-2

The GEOGRAF Data Base

data base structure depends on the application requirements, but within an interactive system it is essential to have a structure that allows rapid access to data elements.

4.3.1 Data Base Items

Within any cartographic data base, two distinct data structures are necessary. As stated in section 4.2, the first structure contains the graphical data sets necessary for the plotting of the map while the second contains data sets that relate feature attributes and descriptions to the physical features on the map. Appendix One contains a description of the data set elements for the NZLS. The survey data on New Zealand cartographic maps comprises features that are made up of:

- (a) straight lines between two points,
- (b) arcs lying on the circumference of a circle,
- (c) irregular curves representing coastlines and rivers.

In section 4.3.3, we explain the techniques and structures used to hold the above items.

4.3.2 Specific Data Base Requirements

The major design criteria affecting the structures outlined in section 4.3.3 are:

- (a) The graphical data must be structured so that the display of a map, or portions thereof, is fast enough to allow generation during interactive digitising, and so that the plotting algorithm is efficient and not subject to error,
- (b) Insertion of graphical data must be simple without the need for any form of pre-processing prior to digitising, and the structure must provide the

ability to associate descriptive data files with their co-ordinate data,

- (c) Modification of the structure through editing and updating should be accomplished with ease,
- (d) The structure should allow additional data sets to be incorporated with minimal difficulty,
- (e) Storage requirements should be minimised subject to the above requirements. For example, we should not store multiple copies of the same co-ordinate.

4.3.3 Detailed Design

In the data base there are two distinct file types; the Graphical Data Set (GDS) and the Descriptive Data Set (DDS).

(A) The Graphical Data Set contains all the data necessary for the reproduction of the three basic types of linework (Section 4.3.1). Features such as coastlines cannot be specified by a single mathematical curve fitting approximation, and are held, in chain encoded form, in a data structure called a CHAIN DICTIONARY. Straight lines and circular arcs are held in a coded format in a single file which is called an XY DICTIONARY.

The XY DICTIONARY contains x and y co-ordinates relative to the bottom left hand corner of the map. Each co-ordinate pair is coded in such a way that, in a single pass through the DICTIONARY, the plotting algorithm can determine whether the current co-ordinate pair represents:

- (a) a point connected by a straight line to the next point,
- (b) the end of a straight line and:
 - i) the start of a circular arc,
 - ii) the next point is the start of a circular arc,

- iii) the next point is the start of new line,
- (c) the end of a circular arc and:
- i) the start of another arc,
 - ii) the start of a line,
 - iii) the next point is the start of a circular arc,
 - iv) the next point is the start of a new line,
- (d) a point that has been deleted from the data base (see Section 5.2.3).

In the coding scheme, a line is defined by two consecutive co-ordinate pairs. Circular arcs are defined by a starting point, any point on the circumference of the arc, and the end point. The plotting algorithm uses the three points to calculate the centre and radius of the arc, as well as determining the starting point in order to generate an arc through the three points. The circular arc algorithm draws anticlockwise and must therefore establish from which point (the first or third) it must commence drawing in order to pass through the second point. This apparently trivial problem is, in many other systems, avoided by restricting the order in which points are digitised, that is, they must be digitised in a clockwise or anticlockwise order.

The coding scheme used in the XY DICTIONARY uses the four possible sign combinations available from two variables. The table on the next page summarises the coding structure interpretation. The algorithm to interpret the coding structure is given in Figure 4-6.

The second structure, the CHAIN DICTIONARY, is used to hold all features composed of irregular curves. Chain encoding was developed by Freeman (1962) and aids in the compression of large data files used for sequences of x and y

Previous Point		Current Point		Interpretation of current point
X	Y	X	Y	
?	?	+	+	Line from current point to next point
+	+	-	+	End of a line
?	?	-	-	Current point has been deleted
-	+	-	+	Previous point was the start of a circle
-	+	-	-	Current point is the start of a circle

Table of XY DICTIONARY codes and their interpretation.

co-ordinates that define curves. The curve is approximated by a sequence of vectors of a unit length in one of eight possible directions from the end of the previous point. Consequently, given a starting point, a scale defining the unit length, and a series of chain codes, it is possible to reproduce the curve.

As the chain length decreases, the accuracy of the approximation increases, but there is a corresponding increase in the number of chain codes. An optimal chain length will depend on the requirements of a specific application. With a smoothing algorithm (see Section 5.3.3), it is possible to increase the chain length and still produce an adequate cartographic reproduction of the feature. Chapter 7 describes a technique developed by the author that further reduces the amount of information necessary to reproduce a curve and allows the user to specify the minimum change in direction from the previous point.

As each direction can be specified by a 3-bit code, it

is possible to pack several codes into a word. Figure 4-3 indicates the structure of the CHAIN DICTIONARY.

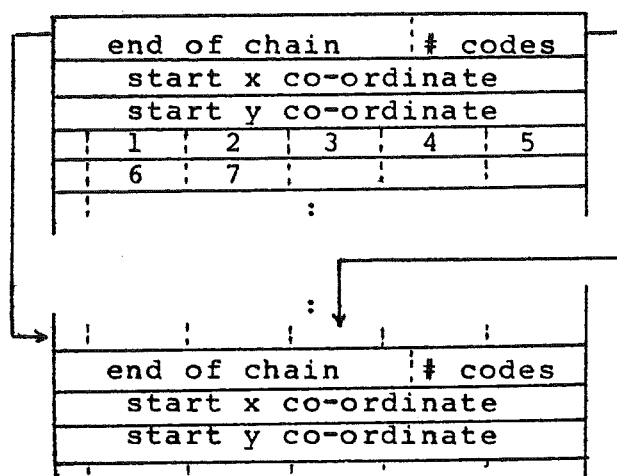


Figure 4-3

Structure of the CHAIN DICTIONARY

(B) The Descriptive Data Set provides two primary facilities. It contains:

- (a) the description of a feature's non-positional data, for example, a parcel's plan and section number, area and C/T reference,
- (b) the definition of features as independent entities by containing references to the co-ordinates in the XY and CHAIN DICTIONARY's that define the feature.

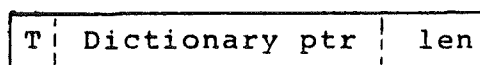
With respect to (a), the data base structure shown in Figure 4-5 allows for both hierarchical and network-oriented relationships to be contained in the data base. At each logical level of the hierarchy, there is a 1:1 relationship with horizontal elements while there is a 1:many relationship between successive vertical levels. For example, every plan number has a 1:1 correspondence with its plot data information but it has a 1:many relationship with parcels. Both the internal and external files are organised as information sets within parallel tables and files respectively. To incorporate

another data base primitive at a horizontal level merely requires the addition of another table which is held in its own external file.

In the event of non linear relationships at the same level, such as a GAZETTE reference, two options are available:

- i) to reserve storage for a 1:1 relationship,
- ii) to organise the structure via indices to provide the logical relationships.

The latter is preferable providing the relationship can be determined without the storage of pointers. MAPPK reserves a bit for every land parcel and sets the bit if the parcel has a GAZETTE reference. Provided that the GAZETTE file is in the same order as the PARCELS file this relationship can be created. A similar technique is used for a 1:many relationship between vertical levels. For example, every feature is defined by several points. MAPPK reserves a TOGGLE bit in the POINT's word to indicate if the current POINT word is the beginning of the definition for another feature (see Figure 4-4). Once again this implies an ordering within files.



where

T : toggle bit
 len : if 0, then ptr references CHAIN DICTIONARY
 > 0, indicates the number of consecutive
 XY DICTIONARY codes for the feature.

Figure 4-4.

Format of a POINT File word

Another technique to reduce storage requirements relies on the ability to determine accurately the visual centre of a geographical feature. If this is possible, the data base need

not hold co-ordinate data to define centroids at which the text related to a feature is to be plotted. However, for text for:

- i) plan types and numbers,
- ii) street names,
- iii) hydrographical features,

it is preferable to hold positional data due to the nature of the feature. In general, it is difficult to calculate a suitable location to plot text data for these features. For example, for i) above, the difficulty in determining the centre of several features leads to the storing of the positional data. A plan number refers to several parcels and within the DDS we hold basic positional data at the parcel level, thus, the total plan area is not a quickly identifiable unit.

It is essential that any system should provide the ability to select and display particular features. This requires correlating cartographic features with their boundary co-ordinates. The selection capability must allow for:

- i) specific physical features to be displayed, for example, roads only, or
- ii) the display of physical features that fulfil some user defined logical requirement, for example, to display all parcels of a specified plan type that are within both a specified area range and a valuation range.

To provide this facility requires an association between each logical feature in the DDS and its corresponding co-ordinates in the GDS. A structure called the POINTS file provides this association by correlating the feature with pointers to its co-ordinates.

Thus, to display a complete map requires only the XY and CHAIN DICTIONARIES while selective displays require the additional feature data as well as their POINTS data. The codes within the XY DICTIONARY still apply when drawing from the POINTS data file. For example, in Figure 5-4, the street is defined by points 11-19 inclusive but the circle codes for 12-18 are determined from the XY DICTIONARY's coding scheme.

In summary, the proposed data structure appears to meet the objectives outlined in the requirements for a data base, as well as overcoming the problems of sliver lines, double lines, plotting efficiency and multiple graphical data types.

Overview of the MAPPAK Data Base

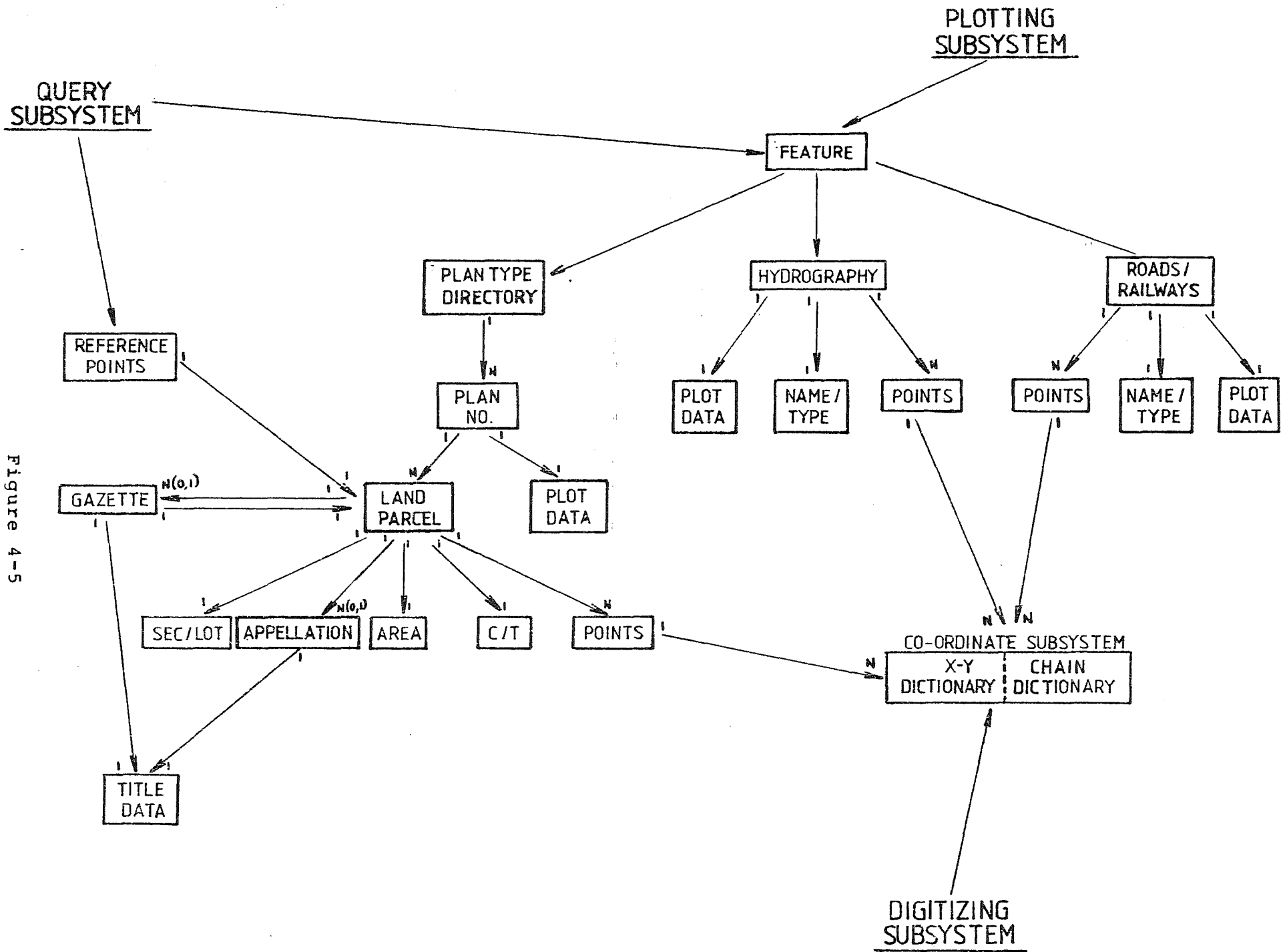


Figure 4-5

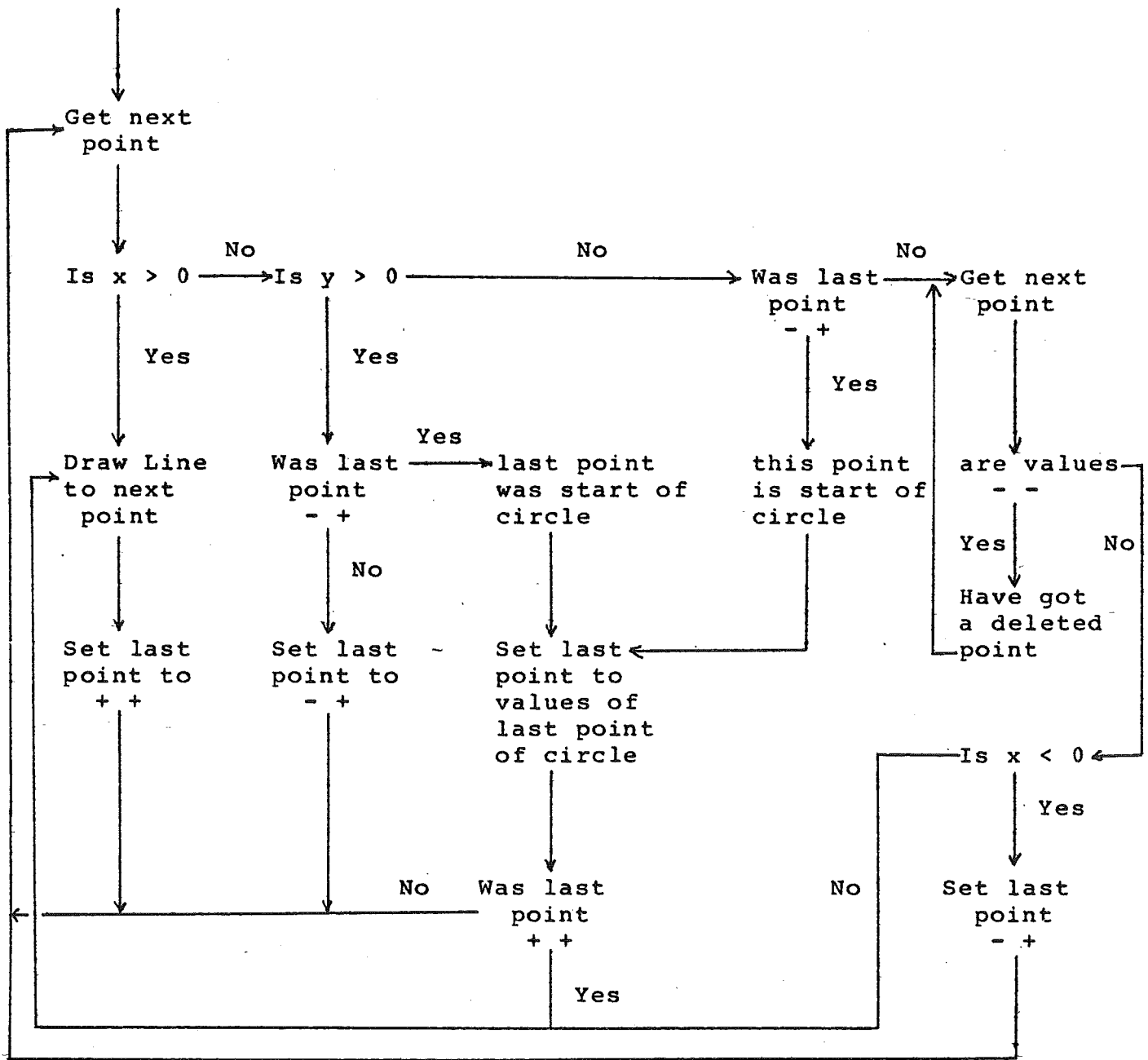


Figure 4-6

A flow chart of the drawing logic.

CHAPTER FIVE

AN INTERACTIVE SYSTEM

Williams (1971) gives the following definition of an interactive system:

" The purpose of an interactive system is to make efficient use of both man and machine allowing the designer to interact with the computer while the program is still running in order to influence the course of his computations."

He implies that the user can exploit the hardware resources more efficiently if he can monitor the progress of his operations in real-time, with backtrack facilities provided by the software. He will consequently require less time to edit the data at a later stage.

Such a system is interpretive in that the operator identifies his intended course of action to the machine, then proceeds to pass data to the machine for verification and processing. The software system must continually verify the syntax and semantics of his commands and, on error detection, inform the operator and indicate the possible recovery procedures. As well as the machine's validation of the input data, interactive systems allow the user to monitor visually the machine's execution of his commands. With a mapping system, the user has the ability to view the map on a graphics display terminal and detect and correct errors in real-time. Any piece of digital data requiring a change prior to producing the final document represents an error.

5.1 INTERACTIVE DIGITISING

An essential component of a computer mapping system is the means to convert raw image data to machine-readable files. At this stage, designers are faced with the problems of retaining the degree of accuracy necessary to satisfy the user's requirements, while operating within the constraints imposed by the hardware. The success of a cadastral mapping system depends on this graphical to digital conversion.

Manual digitisation can produce accurate input data, but it is a slow, tedious process subject to many types of error. The user requires many aids to reduce fatigue and errors. The system must provide the user with the means to recognise and correct errors, preferably while the map is still oriented on the digitising table. Many of the errors that occur can be detected only by visual inspection. Very few errors can be detected automatically by software, and those that can require complicated and time consuming procedures to rectify them.

Interactive digitising is economical in that error detection and correction are performed during data acquisition. If the system's design does not cater sufficiently for these functions, then a costly automated digitising process may become an economic alternative.

5.1.1 Advantages

The most notable advantages offered by interactive digitising compared with either off-line digitising or automated scanning are:

- (a) the hardware required is cheaper than that of a wholly automated system,
- (b) minimum operator training is required. With a

question-and-answer type approach, detailed instructions can be given through an operator's display,

- (c) early error detection and subsequent correction is made while the map is still oriented,
- (d) interactive digitising aids in the generation of a base map from multiple sources as the matching process can be monitored visually.

The digitising process is simplified in that no entity is identified at the time of digitising. This function is left to a later stage when the graphical data is combined with its descriptive information. Also, provided the correct transformation matrices are used for transforming each digitised point, the source documents need not be aligned with the digitiser axes.

5.1.2 Menu Driven Digitising

A set of menu commands in a reserved area on the digitising table provides the user with an effective means of identifying the parameters for the interactive digitising software. This technique is preferable to that of manually entering the input parameters through an operator console, provided the interpretive language can be implemented by simple menu commands.

Use of a menu requires less system resources and results in higher throughput as the operator does not continually alternate between input devices. A table of menu commands is given on the digitiser. Whenever the software detects a digitised point in the menu area, this point is mapped to its corresponding code. The input commands are analysed to verify that the instructions are semantically correct given the present state of the system. On error

detection, the system must identify the error and provide the user with a set of possible recovery options.

(A) Menu Commands: MAPPAK uses a table-driven interpreter to execute menu commands. Because of the interpreter's tabular nature, it can readily be modified to cater for additions to the instruction set. Each menu pick is converted to an index into a table where the corresponding table attribute defines the nature of the command. The menu commands on the digitising table are:

- (a) DELETE POINT: requests that the previous point digitised be deleted if possible. If the previous point resulted in a line being drawn on the interactive terminal, then the current picture must be redrawn to restore the display. This is an error correction mechanism.
- (b) DELETE INSTRUCTION: requires all points associated with the current processing mode, that is, line, circle, or stream mode, to be deleted. This command has the same effect as a DELETE POINT, except that it performs a multiple deletion in one operation. It is useful where an error is not detected until some time after its occurrence.
- (c) INCLUDE PREVIOUS POINT: generates a duplicate input of the previous point digitised. This command is useful when a digitised point causes the current screen picture to be redrawn (see Section 5.1.4). While in the redraw state, the cursor may be removed from a point that requires digitising twice. This function then generates a pseudo input of the same point on redraw completion.
- (d) LINE: sets the current processing state to line

mode. Entering line mode using this pick indicates that the previous point digitised is also the start of a line. It is used when the last point of an arc or stream also represents the start of a straight line feature.

- (e) NEWPOINT LINE: is the same as LINE except that the next point digitised is the first point for this mode.
- (f) ARC: sets the current processing state to circular arc mode. Entering arc mode implies that the last point of the previous mode, plus the next two or more points, will be taken to define an arc. More than two points will be accepted. In this case, a multiple number of arcs are defined, in which the last point of an arc is also the first point of another arc.
- (g) NEWPOINT ARC: same as ARC mode except that the next point digitised represents the first point for an arc.
- (h) STREAM: sets the current processing state to stream mode and implies that the previous point digitised was the start of stream mode. In this mode, points are continuously captured while the operator traces the outline of a feature with the cursor.
- (i) NEWPOINT STREAM: same as STREAM except that the next point defines the start of stream mode.
- (j) BLOCK: indicates that the co-ordinates input next define a block boundary. Such co-ordinates are held in the XY DICTIONARY with suitable identifiers in the Information Header Block (see Appendix Two)

to point to the start and end of the block co-ordinates in the XY DICTIONARY. This is an application-dependent feature.

(k) TITLE: indicates that the following co-ordinates represent centroids for the plotting of text data not related to individual parcels, for example, plan numbers, and road and railway names.

(l) END: indicates the end of a digitising session.

This set of menu commands can be expanded by defining the new menu items and including them in the appropriate place in the table.

(B) Command Semantics: the interpretation of a menu command depends primarily on the current digitising mode and the state within this mode. Possible modes are BLOCK, LINE, ARC, STREAM and TITLE. Also, a NULL mode is entered following the DELETE INSTRUCTION command. Within each mode there are various states. In arc mode, for example, the 4 states are:

- (a) no co-ordinates digitised,
- (b) one co-ordinate defined,
- (c) two co-ordinates defined,
- (d) three co-ordinates representing the arc are defined. This represents an end state in this mode.

Thus, if in states (b) or (c) in arc mode, a request to change mode is rejected as we have not reached a completion state within this mode.

Consequently, many types of error are possible. On error detection, the corresponding error must be identified to allow the operator to make the correction. Initial identification of an error must interrupt the operator. One technique is to generate an audible signal and display the

error code on the terminal. Two possible approaches are:

- (a) An error message could be displayed. With a storage tube, this technique requires that the current map frame being displayed be redrawn when the error message area fills, as these screens do not have a selective erase capability.
- (b) An area containing a set of predefined error messages can be permanently displayed on the screen. On error detection, some marker is set against the error message. With a storage tube, the cursor may be used as it can alter position without the screen being redrawn.

Rather than giving error messages, MAPPAK specifies the possible recovery actions. This assists the operator, as well as reducing the storage requirements for a reserved error message area. The semantics checking identifies the following possible corrective actions:

- (a) DELETE INSTRUCTION: for example, if in stream mode an attempt to delete a point is rejected, but it is possible to perform a DELETE INSTRUCTION command.
- (b) DELETE POINT/INSTRUCTION: for example, in circle mode in states (b) or (c) above, and an attempt is made to exit to another mode.
- (c) PICK MENU AGAIN: for example, in NULL mode and a map co-ordinate is digitised, or an unidentifiable menu pick is made.

An informative message CANNOT DELETE POINT is used when an attempt is made to delete a point when either in the NULL mode, or successive delete point commands return the system to the initial state for the mode.

5.1.3 Map Orientation

Many references (for example Tomlinson 1977) describe the common problem of map orientation and the need to align the map with the x and y digitiser axes before digitising can proceed. Baxter (1976) provides an introduction to the use of affine transformations. According to the theorem of plane perspective, four points in one co-ordinate system and four points in another system completely define a projective transformation. Therefore, four co-ordinates defining a map's corner points in the digitiser's co-ordinate system are sufficient for a transformation of any digitiser co-ordinate to another co-ordinate system. The new co-ordinate system could be the actual map grid co-ordinates, or a system in which each co-ordinate is specified relative to a base point on the map, most logically the south west corner point. The latter system is preferable in that not only can the actual map grid co-ordinates be calculated but, more important, the same transformation is applied to every map and the magnitudes of the transformed co-ordinates are predefined. This has the added advantage of requiring only single precision arithmetic on a 16-bit minicomputer.

Appendix Three describes the two transformation routines. One routine is needed to define the affine transformation initially, while the second routine is a matrix transformation of digitiser co-ordinates to the new co-ordinate system.

The properties of an affine transformation are such that within two frames of reference, straight lines remain straight, parallel lines remain parallel, and the ratio of a point dividing a line in the original system is the same as that for the transformed system. Therefore, no alignment

before digitising is required as this is performed by the transformation. For interactive systems that display maps as they are digitised, a transformation routine is essential for generating accurate displays.

In MAPPAK, the four map co-ordinates are transformed to (0,0), (0,5000), (5000,7500), and (7500,0) respectively. This transforms any map co-ordinate to a .1mm resolution in relation to a 50cm x 75cm map sheet. A windowing scheme is therefore not required when plotting maps as the co-ordinates can be used directly by the plotter.

When a point is digitised, it is tested to determine whether or not it is a menu point; all menu points have an x co-ordinate value less than a predefined x value on the digitising table. If the point is not a pick of a menu command, it is transformed to the new system and tested against the current window (Section 5.1.4). A new map frame is drawn when digitising points that are not in the display frame currently on the screen.

Rogers (1976) gives a general 3x3 transformation matrix for 2-dimensional homogenous transformations:

$$\begin{bmatrix} a & b & p \\ c & d & q \\ m & n & s \end{bmatrix}$$

where a,b,c,d affect scaling, sheer and rotation, s affects overall scaling, m and n are translation factors and p and q provide a projection. Using such a transformation, modification from one co-ordinate system, namely the former imperial mapping grid, to another system can be implemented by defining suitable transformation parameters.

5.1.4 Windowing

A display screen is generally too small to provide a complete picture of the source document at the final output

scale. However, if the interactive system is to provide error detection, we should generate a display of the map at the final scale.

Consequently, a windowing scheme is necessary in which portions of the map are visible at the final scale. If the operator continually monitors his progress visually, the changing window and his inability to view the complete document will not restrict the efficiency of the system. Success of this technique depends on the ability to regenerate rapidly the current contents of a new window. Fast picture regeneration depends on the data base and the ability to extract information from the coding structures.

Two types of windowing scheme were considered; fixed windows and variable windows. Fixed windows imply pre-defined window co-ordinates, an example being the division of a 50cm x 75cm map into six 25cm x 25cm windows or map frames. A variable window frame would have the same dimensions as a fixed, but variable co-ordinates for its corner points. An example is a technique that uses, as the midpoint of the new display frame, the digitised point that causes a change in the display frame.

Experiments suggested a variable window would be preferable to a fixed window. When a map is densely digitised, with both XY DICTIONARY data and CHAIN data, long delays may be experienced during changing the display frame due to the 'clipping' procedure (see below). The aim is therefore to minimise the total number of frame changes required. In a fixed window scheme, it is not unusual for successive digitised points to alternate between windows.

Another type of windowing is also necessary for redrawing the contents within the display frame. Features

that fall outside the display frame must not be displayed. Also, a feature that is bisected by the display frame must have the linework outside the frame 'clipped'. Most graphics software packages provide standard routines to aid the definition of 'virtual windows', which result in automatic 'clipping'. The objective is to provide the programmer with the tools to map a variable sized picture frame to a fixed area of the screen. The overall result, a scaling technique, is achieved by redefining the picture frame to be displayed.

MAPPAK uses a variable window technique with six 25cm x 25cm frames that give real-time displays at a scale adequate for error detection. To decrease the picture regeneration time it was necessary to store chain encoded features initially as raw x y co-ordinates to remove the overhead in decoding the data. On completion of the digitising process, the raw data is converted to chain encoded data and stored in a compact form.

5.1.5 Multimap

When generating metric base maps from the former hand drawn imperial record sheets, we needed to capture data from several imperial sheets to generate one new metric base map. Base map sheet sizes and scales differ between the metric and imperial systems, and data for the new metric base may spread over as many as four imperial sheets. The procedure to capture and merge data from several imperial sheets must maintain the cartographic accuracy of the input source documents. Two possible approaches for this procedure were studied.

The first technique required that the appropriate sheets be digitised individually, and that the internal boundary points (for example, points q_1 and q_4 in Figure 5-1)

be merged after digitising all appropriate sheets. This requires that the four polygons defined by the points

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

be individually digitised, then merged. For METRIC 1 in Figure 5-1, the following mapping for the transformation routine applies:

points	X	Y
1,1	0	0
1,2	L1	0
1,3	?	?
1,4	0	L2

where L1 = ratio of length (1,2 - 1,1)/(2,2 - 1,1) * 5000
and L2 = ratio of length (1,4 - 1,1)/(4,4 - 1,1) * 7500

In the general case, points (1,3), (2,4), (3,1), and (4,2) are not digitised. Instead, suitable reference points are generated each time an imperial map is digitised. For point (1,3), we want to generate a new corner point h (see Figure 5-2) in which (1,3) is then relative to the four map boundary points. The following algorithm is used to determine the actual digitiser table co-ordinates of this new point:

Given that i follows h in an anticlockwise direction, then

$$\begin{aligned} j &= \text{mod}(i, 4) + 1 \\ k &= \text{mod}(j, 4) + 1 \\ h &= \text{mod}(k, 4) + 1 \end{aligned}$$

and the table co-ordinates are

$$\begin{aligned} x' &= X_i - X_j + X_k \\ y' &= Y_i - Y_j + Y_k \end{aligned}$$

The transformation matrix co-ordinates depend on which map we are dealing with. For

metric 2 or 4

$$\begin{aligned} x_h &= x_i \\ y_h &= y_k \end{aligned}$$

metric 1 or 3

$$\begin{aligned} x_h &= x_k \\ y_h &= y_i \end{aligned}$$

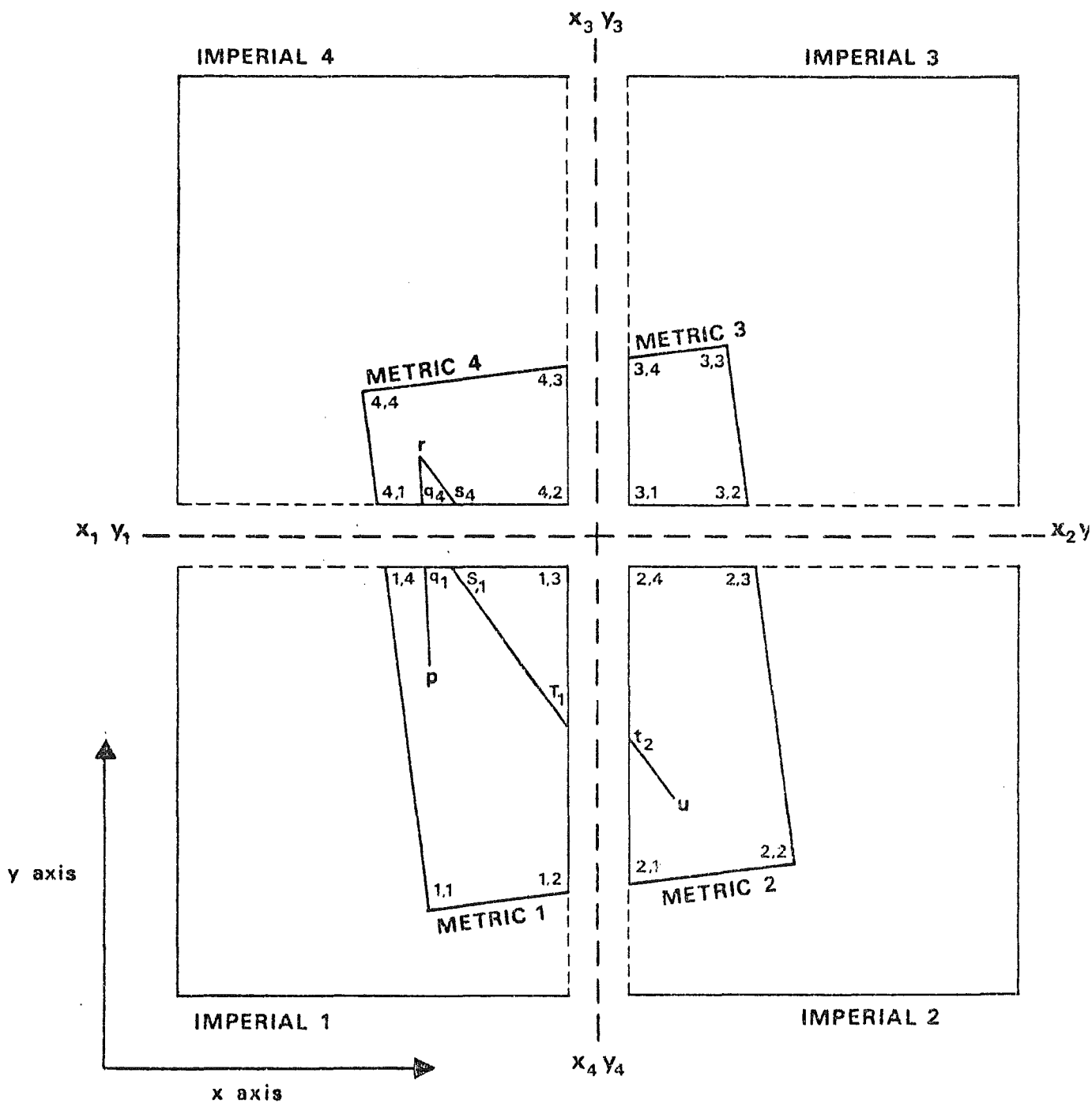


Figure 5-1

Digitiser setup for Multiple Base Maps

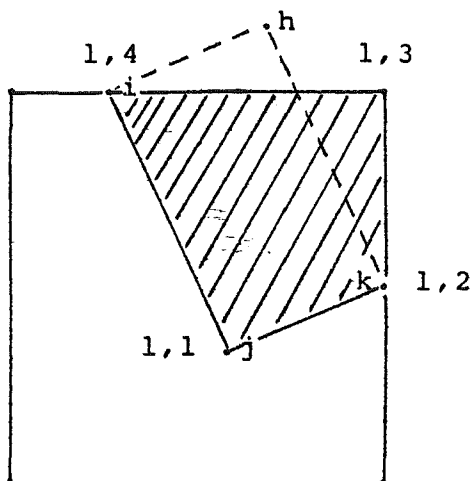


Figure 5-2

Example of generating 4th
co-ordinate for transformation routine

Graphical features that are common to several sheets, such as vector pr , require that the partial vectors be digitised on their separate sheets and internal vector boundaries removed on completion to form a single vector.

Three reasons for rejecting this approach were:

- (a) internal vector co-ordinates will seldom correspond across sheet boundaries when digitised using different transformation matrices, and subsequent matching and elimination of common points is difficult.
- (b) the complex nature of the merging procedure. Consider, for example, the merging of vectors rs_4 , s_1t_1 , and t_2u into the single vector ru .
- (c) the approach does not provide for visual error detection and correction when building a base map from multiple sources.

The second approach is to place all the required imperial maps on the digitiser, to associate each metric map

segment with a separate transformation matrix, and to digitise features that span more than one sheet as though they are on a single sheet. For the vector from r to u , point r would be digitised and transformed using the transformation matrix for METRIC 4, while point u would be transformed by a different transformation matrix. Points s_1 and t_1 are therefore never digitised, thus eliminating one possible source for error. This technique requires that the map portions be placed on the digitiser as shown in Figure 5-1, but not necessarily parallel to one another. The procedure is then to:

- (a) identify to the system the number of sheets to be digitised,
- (b) input points (X_1, Y_1) and (X_2, Y_2) , then points (X_3, Y_3) and (X_4, Y_4) . Each pair of co-ordinates defines a line and, from these, the software can determine to which map segment a digitised point belongs.
- (c) digitise all metric map boundary points in anticlockwise order, except for points (1,3), (2,4), (3,1) and (4,2). Metric map corner co-ordinates are identified to allow ratios and subsequent transformation matrix co-ordinates to be calculated for points such as (1,2) and (2,1).
- (d) on completion, the table co-ordinate and transformation co-ordinate matrices will contain all co-ordinate pairs except for internal points. Where three points have been digitised, the fourth is calculated using the procedure described for calculating h in Figure 5-2. When only two points exist, they are used to calculate another two such that the four points generate a square. All

co-ordinates are now defined for the transformation routines.

Having completed the initialisation, digitising proceeds in the same manner as for a single sheet.

5.1.6 Digitising Sequence

Several factors contributing to the system's efficiency are dependent on the digitising sequence. To aid throughput, no pre-processing is required prior to digitising, and no entity is identified during digitising. As a result, the digitising sequence is at the operator's discretion. The order in which points are digitised affects both the efficiency of the plotting algorithm and the storage requirements of a particular map.

The plotting algorithm reproduces the features in the order they were originally digitised, and consequently the aim is to minimise the amount of unproductive plotter head movement. It is uneconomical to digitise features randomly from the source document in preference to incorporating some logical flow throughout the digitising sequence.

Two file structures are affected by the order in which points are digitised; the XY DICTIONARY and the POINTS file used to hold indexes to the XY dictionary for a particular feature (see Appendix One). For the simple example in Figure 5-3, the digitising sequence may be either A,B,E,F and then B,C,D,E, or alternatively A,B,C,D,E,F.

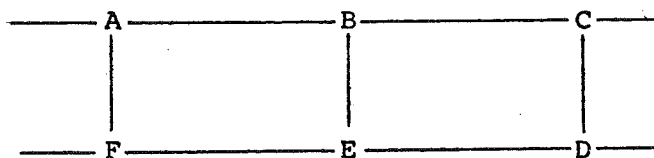


Figure 5-3

Example of Digitising Sequence

In the first instance, duplicate copies of points B and E are held in the XY DICTIONARY which represents a possible source of error. In the second case, due to the non contiguous nature of the points defining both features, the POINTS data for each feature will contain two entries, rather than one required for the first case.

5.1.7 General Digitising Process

After positioning the maps on the digitiser, the digitising procedure issues the following initialisation requests:

- > Enter Map Name: e.g S14-1000/34.12
- > Enter Chain Length: e.g 4 (implies a .4mm chain)
- > Enter Number of Maps: 1-4
- > Digitise 4 Points for Centre Lines of Maps:
- > Digitise Map Boundary Points:
i.e the corner points of map sheets
- > Digitise First Northing and Easting:

The operator then proceeds to digitise the map segments, identifying his actions by the digitiser's menu commands. The mode of operation is continually displayed in the menu area of the interactive display device. After a display has been generated, the cursor repositions itself at the appropriate mode position. Within each mode, the present state is maintained to provide semantics checking for menu commands. Mode identification is necessary only when a change of mode is required. The following attributes apply for each mode:

- (a) LINE: if the current point represents the end of one line and the start of another, the point is digitised twice. The software detects this

condition and records the point only once in the XY DICTIONARY with the appropriate code. Independent lines are digitised as such.

- (b) CIRCLE: requires at least 3 points to identify an arc; the start point, any point on the circumference, and the end point. Multiple arcs may be defined in this mode.
- (c) STREAM: requires a change in the hardware digitising mode to stream mode. The operator then traces the outline of the feature with the cursor. In this mode it is essential to capture and process digitised points quickly. Points are therefore not converted to chain codes until the end of the digitising session, and the cursor is not repositioned each time to indicate the mode.

It is possible to create the XY DICTIONARY, CHAIN DICTIONARY and TITLE files in 3 independent digitising sessions, as these files are independent of each other.

The following example describes the digitising sequence to generate a part of Figure 5-4:

Enter Line mode after digitising point 11 in circle mode
Digitise point 12,
Enter circle mode,
Digitise points 13 14 15 16 17 and 18,
Enter line mode,
Digitise point 19,

In circle mode, the 3 arcs between points 12-14, 14-16, and 16-18, were defined. The following mode defined the line from the end of the last arc to point 19.

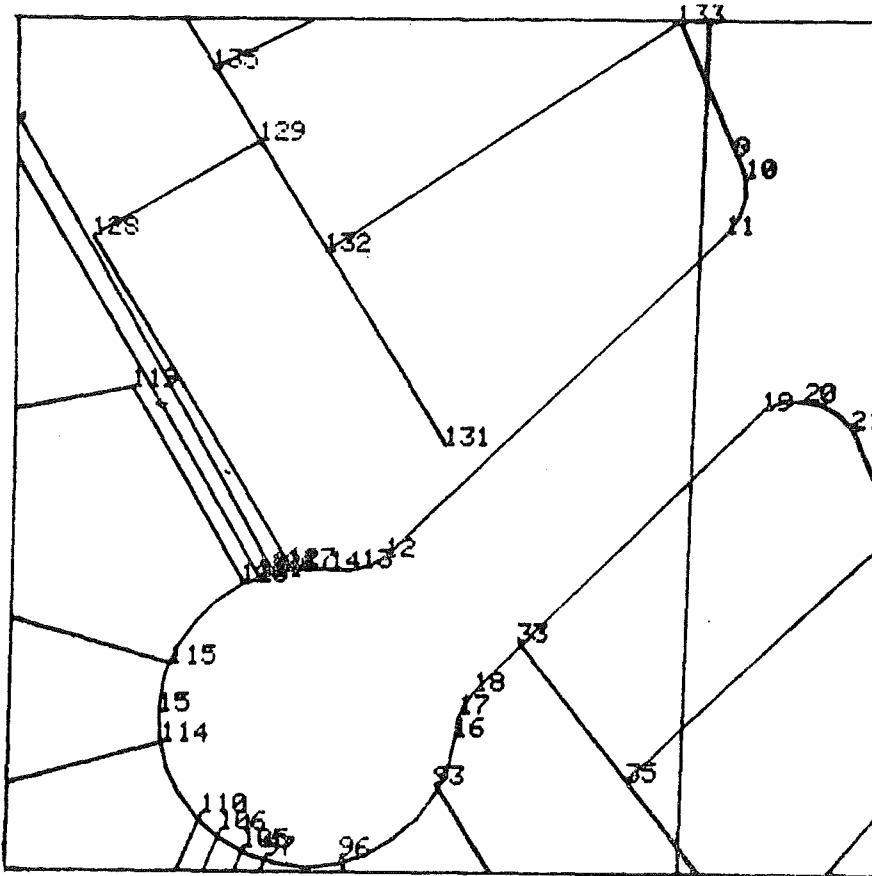


Figure 5-4

Display of an arc on VDU

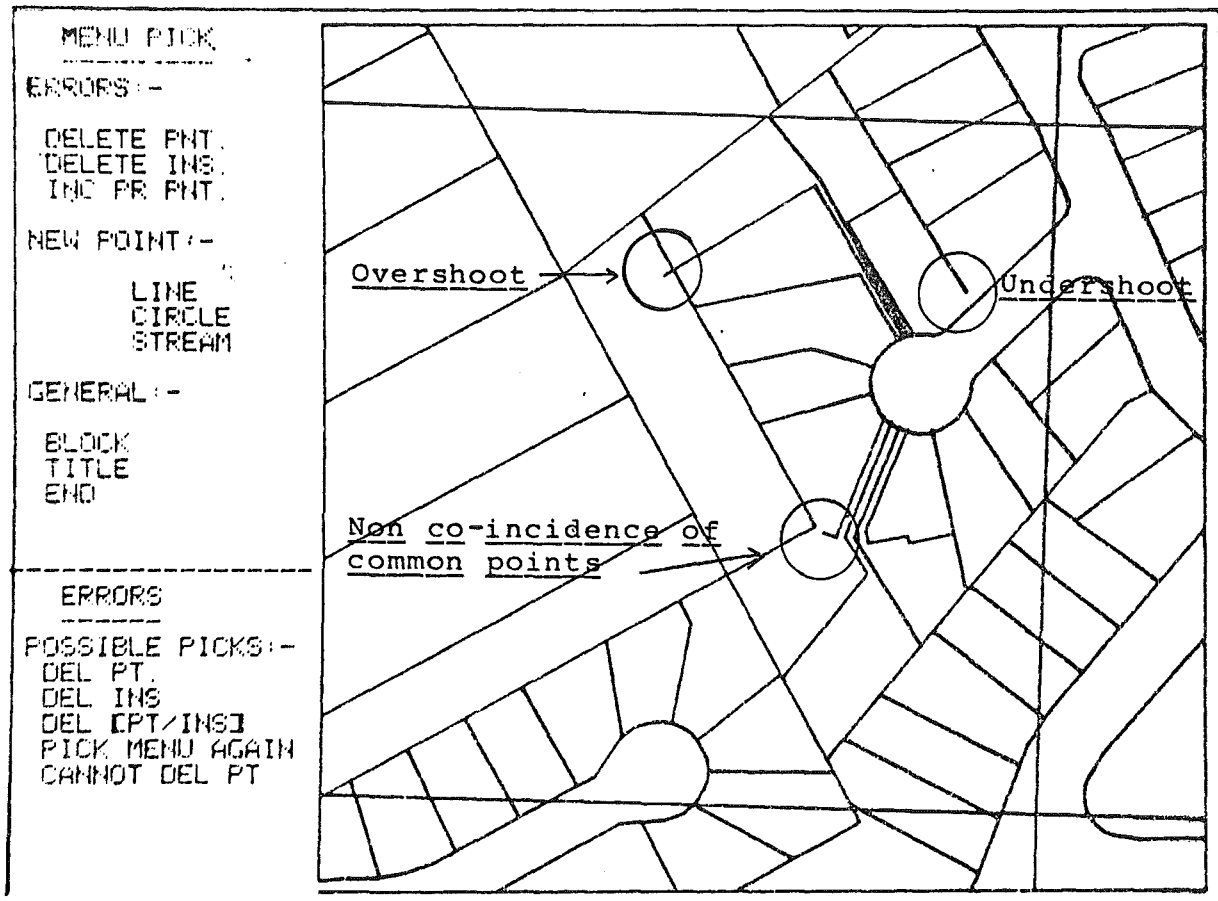


Figure 5-5

Interactive digitising screen format

5.2 EDITING AND UPDATING

In an interactive system it is necessary to cater for correcting errors that are undetected during the digitising phase. Interactive editing has the advantages that:

- (a) on error detection, the operator can execute and monitor the correction process visually,
- (b) it is economical in that error detection occurs as the result of the descriptive data acquisition function.

In a computer mapping system, a GDS and DDS must be created. Interactive digitising provides the GDS. Generation of the DDS requires co-ordinate identifiers from the GDS and data from the original source documents. While the DDS can be created simultaneously with the GDS, a more complicated procedure is required and time on the digitiser is wasted. The author proposes therefore to keep the two processes independent.

During DDS creation the operator will find errors in the GDS, and he must switch from the DDS generation mode to an error correction mode, correct the error, and return to his previous mode. The hardware used by MAPPK is shown in Figure 2-1 where:

- (a) the teletype/VDU provides a keyboard for entering both DDS data and edit commands,
- (b) the display terminal provides user selected windows into the GDS as well as the corresponding point identifiers,
- (c) the digitiser forms an additional source for edit/update input.

5.2.1 Editing

No attempt has been made to correct errors automatically. The system relies wholly on visual inspection for the identification of errors. To compensate automatically for errors such as lines not correctly joining would require very complicated procedures and long computation times. The most economical approach is to have an interactive system. Figure 5-5 shows the most likely types of error, most of which can be detected at the digitising stage. Those errors exclusive to the editing procedure and not catered for by the updating routines are:

<u>ERROR</u>	<u>CORRECTION ROUTINE</u>
1) Under/Overshoot	JOIN P_i, P_m, P_n
2) Non co-incidence of common points	EQUATE P_i, P_j
3) Undefined intersection	GENERATE P_i, P_m

5.2.2 Updating

Here, "updating" is concerned primarily with the techniques for changing the GDS. Changes to the DDS require restructuring the external files when adding or deleting parcel data, or modifying existing data for changes in parcel attributes. Additions and deletions require the update process to modify memory resident chain structures depicting the order of the records. On completion of the DDS update, the internal files are written back to the external files in the order indicated by the chains.

The GDS update functions must cater for:

<u>FUNCTION</u>	<u>PROCEDURES</u>
1) Addition	ADD, CONNECT P_i, P_j , and any EDITING routines
2) Deletion	
i) lines	DELETE LINES P_i, P_j, \dots, P_n


```

<parcel list> ::= <parcel details> <parcel list> |
                 <parcel details> <control code>

<parcel details> ::= <parcel> | <parcel> <Gazette>

<parcel> ::= [lot/sec #] [area] [C/T] [<point list>]

<Gazette> ::= [year] [page #] [purpose]

<point list> ::= <point id> | <point id> <point list>

<point id> ::= point | point-point | Stream point

<Road/railway> ::= <feature details>

<Hydrography> ::= <feature details>

<feature details> ::= [mapid] [name] [<points>] <control code>

<escape code> ::= <edit/update fn> <control code>

<edit/update fn> ::= UPDATE
                    CONNECT~Pi~Pj
                    DELETE <feature type> <point list>
                    EQUATE~Pi~Pj
                    GENERATE~Pi~Pm
                    JOIN~Pi~Pm~Pn
                    PRINT <display type>
                    SUPRESS <display type>
                    SCALE~Pi~scalefactor
                    WINDOW~mapframe

<feature type> ::= Line | Arc | Chain

<display type> ::= Co-ordinate identifiers | Title points |
                  Chains

```

B) Semantics

a) UPDATE. It is essential to cater for incorporating new survey data in existing GDS files. As Larsson (1975) said,

"No cadastre will serve its intended purpose unless the currency of data is maintained through a continuous updating process"

New data may be entered either directly from the surveyor's field notes or from a manually updated copy of the source document. The digitiser is used to append data to existing GDS files. The manually updated version of the map is placed on the digitiser and the four corner points are digitised to

establish orientation for the transformation routines. The UPDATE process uses the full set of digitising commands.

As a result of the XY DICTIONARY coding scheme and GDS file structures, data can be easily appended to both the XY and CHAIN DICTIONARY. The user monitors his update process on the display screen. Provided the four corner points are digitised correctly, subsequent digitised data will be accurately merged with existing GDS data.

To update from surveyor's field notes requires collation of the co-ordinate data with its corresponding descriptors, that is, line, arc, or chain, and passing it through an update process. Using the South West base point in the data base, the data entered can then be transformed to fit the new reference system.

(b) CONNECT $P_i \sim P_j$: This function connects the two existing points P_i and P_j by a straight line. The two points are duplicated and added to the end of the XY DICTIONARY with the appropriate codes.

(c) DELETE: Deletions are performed by manipulating the codes in the XY DICTIONARY rather than by physically removing the co-ordinates. This technique prevents altering the position of following co-ordinates in the XY DICTIONARY, and thus destroying the associations maintained in the parcel identifier files. Fragmentation results in that unused co-ordinates are held in the XY DICTIONARY. However, such locations are threaded together, and these 'free' locations are used by the update process.

Input of:

- i) DELETE~LINES~ $P_i \sim P_j \sim \dots \sim P_n$ causes the straight line between P_m and P_{m+1} to be removed. The x and y co-ordinate codes are modified so that codes for

points P_{m-1} and P_m are -ve -ve.

ii) DELETE~ARCS~ P_n causes the arc defined by points P_n , P_{n+1} and P_{n+2} to be removed. If the first point on the arc, P_n , is not related to point P_{n-1} , then P_{n-1} , P_n and P_{n+1} must have their codes changed to -ve -ve, otherwise, only points P_n and P_{n+1} must be modified.

iii) DELETE~CHAIN~ P_n causes the complete set of chain codes related to the P_n identifier to be deleted.

Figure 5-6 is an example of how the coding scheme is used to remove the arc defined by points 14, 15, and 16 in Figure 5-4.

Before change		After change	
Codes		Codes	
	x y	x y	
11	+ +	+ +	
12	- +	- +	
13	- +	- +	
14	- +	- -	
15	- +	- -	
16	- +	- +	
17	- +	- +	
18	+ +	+ +	
19	+ -	+ -	

Figure 5-6

Example of deleting arc defined by points 14, 15, and 16 in Figure 5-4.

(d) EQUATE $P_i \sim P_j$: This function sets the co-ordinates of point P_i equal in value to those of point P_j . To correct

the mismatch of points 54 and 112 in Figure 5-7 we would specify EQUATE 112 54.

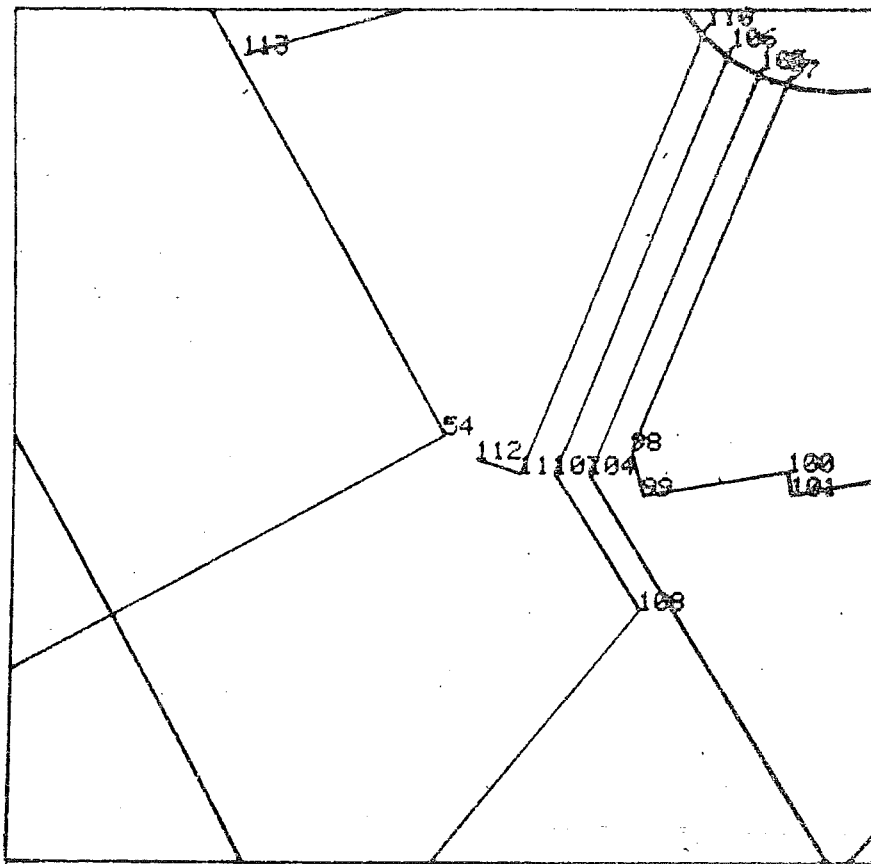


Figure 5-7

Non coincidence of points 54 and 112

(e) GENERATE $P_i \sim P_m$: This generates a point P_k such that the line between points P_i and P_j intersects the line between points P_m and P_n at point P_k . Such a point is necessary when establishing a parcel's identifiers in which point P_m was not digitised. Points P_j and P_n are calculated in the same manner as point P_j in the JOIN procedure.

(f) JOIN $P_i \sim P_m \sim P_n$: This function corrects over and/or undershoot. (See Figure 5-5). A new co-ordinate value for P_i is calculated so that P_i lies on the line defined by points P_m and P_n . Point P_j is obtained on the basis that if P_i 's codes are +ve +ve then P_j equals P_{i+1} , otherwise P_j equals P_{i-1} . The new value of P_i is calculated by solving the two sets of

simultaneous equations of the form $Ax+By=C$ defined by the four points.

The other routines defined in the syntax for editing and updating are auxiliary routines for defining windows, scales and display features, and are of primary importance in the creation of the DDS files.

5.3 PLOTTING SUBSYSTEM

An important factor contributing to the success of a computer mapping system is that the computer generate maps that are not noticeably different from manually drafted documents. Computer produced documents can be evaluated in terms of accuracy and quality. In this thesis, "accuracy" refers to the difference between a computer produced document and the original source document. This difference can be checked through visual inspection of a computer produced document on transparent foil, superimposed on the original document. This initially assists in detecting any plotting algorithm errors. "Quality" refers to the legibility of the document, the location and presentation of text, and the overall aesthetic appearance.

5.3.1 Introduction

The plotting subsystem is to produce maps on any type of graphical display device, in particular on both display tubes and plotters. Chapter Four describes the general technique for processing the XY DICTIONARY codes. The additional aspects concerning the location and style of text, well constructed arcs, and smooth chain reproduction, require techniques that add to the display generation time. For interactive work, a reduction in quality permits faster

display generation. Provided that the display is good enough for error detection, the aesthetic requirements are lower than those for the final document. This section describes the techniques used to incorporate the aspects of map making other than linework, and how these techniques provide the quality needed for a cartographic document.

5.3.2 Arc Generation

Circular arc generation is necessary for both feature reproduction and text generation. An algorithm to produce an accurate and smooth arc is complicated, requiring many calculations. The quality of the arc is therefore a function of the map's use. Picture generation time is more important than picture aesthetics in an interactive error detection environment, but, the reverse is true when producing the final hardcopy document.

An arc generation procedure requires processing time to calculate:

- (a) the centre of the arc from three circumference points,
- (b) whether to draw the arc clockwise or anticlockwise in order for the arc to pass through all three circumference points,
- (c) the number of straight line segments to approximate the arc, and, for each segment, its corresponding end points.

The arc centre is necessary for the plotting procedure. The three circumference co-ordinates substituted in the formula

$$(X-X_c)^2 + (Y-Y_c)^2 = R^2$$

where (X_c, Y_c) are the co-ordinates for the centre of the circle and R is the radius, give three simultaneous equations

involving three unknowns X_C , Y_C , and R . The general equation is:

$$X_i^2 - 2X_iX_C + X_C^2 + Y_i^2 - 2Y_iY_C + Y_C^2 = R^2 \quad \text{for } i=1,2,3$$

Substituting for R^2 in two equations and factorising the resulting expression gives two simultaneous equations:

$$\begin{aligned} 2(X_2 - X_1)X_C + 2(Y_2 - Y_1)Y_C &= X_1^2 + Y_1^2 - X_2^2 - Y_2^2 \\ 2(X_3 - X_2)X_C + 2(Y_3 - Y_2)Y_C &= X_2^2 + Y_2^2 - X_3^2 - Y_3^2 \end{aligned}$$

which are in the form $Ax + By = C$. This form of the expression has eliminated the non-linear factor present in the original expression and can therefore be solved directly.

The decision concerning the direction of arc generation is substituted for one that determines the starting co-ordinate for an anticlockwise drawing algorithm. During digitising, the order in which arc co-ordinates are to be digitised is not fixed. Therefore, unless the correct starting point is used, an incorrect arc will be generated. An algorithm for determining the starting point is as follows:

- (a) let (X_C, Y_C) be the origin of a set of XY axes, and let (X_P, Y_P) be a point on the X axis in the first quadrant,
- (b) for each circumference co-ordinate (X_i, Y_i) , calculate the angle θ between (X_i, Y_i) and (X_P, Y_P) with (X_C, Y_C) as the vertex,
- (c) set i to 1, 2, or 3 depending on the co-ordinate that generates the smallest angle θ_i ,
- (d) if angle $\theta((i+1) \bmod 3) < \text{angle } \theta(((i+1) \bmod 3)+1)$ then
 the starting co-ordinate is the 3rd arc point
 else
 the starting co-ordinate is the 1st arc point

The total angle subtended by the arc is used to calculate the number of segments to approximate the arc. Too few segments

result in a crude circular arc approximation but a fast generation time, whereas a large number of segments produces a smoother curve at the expense of generation time. The smoothness required is a qualitative judgement and depends on the user's requirements. One function for determining the number of segments uses both the total angle and the radius. As the radius increases, so do the number of segments. A function in the form:

$$\text{angle of the arc} * R^2 / \text{CONSTANT}$$

where CONSTANT, which is application dependent, provides a suitable means for generating the number of segments. The value of CONSTANT will depend on the resolution of the display device. In the NZLS system, CONSTANT was set to 60 to provide suitable displays on the plotter.

Having calculated the total angle and number of segments to approximate the arc, the following algorithm for generating successive circumference co-ordinates is used:

(a) let θ be the total arc angle divided by the number of segments to approximate the arc, and calculate $\text{SIN}(\theta)$ and $\text{COS}(\theta)$;

(b) set XPL and YPL to the starting x and y co-ordinates respectively;

(c) for each segment on the arc, calculate its x and y end points as follows:

$$XP = X + XPL * \text{COS}(\theta) - PYL * \text{SIN}(\theta)$$

$$YP = Y + PXL * \text{SIN}(\theta) + YPL * \text{COS}(\theta)$$

and draw the line segment between the two co-ordinates (XPL,YPL) and (XP,YP);

(d) set XPL = XP, YPL = YP, and repeat (c).

This method of arc construction is used wherever the XY DICTIONARY contains arc codes. It attempts to reduce the

amount of compute time required to process arcs, and allows the user to control the smoothness of the arc.

5.3.3 Chain Encoding/Decoding

Chapter Four introduced the concept of chain encoding (Freeman 1962) and its usage. A 3-bit chain code resulting in 8 possible directions was described. During digitising, raw XY co-ordinates are continuously captured for features whose points lie along curved lines, and saved in their raw form. This method permits rapid capture, processing and storage of incoming co-ordinates, and removes the overhead of translating chain encoded data during digitising. On completion of the digitising session, the co-ordinates are converted to chain codes to reduce their storage requirements. An algorithm for converting co-ordinates to Freeman chains is given in Appendix Four. Results have shown that a file containing 50 records of raw chain data can be reduced to 10 records when using a chain length of .4mm.

The post-digitising plotting algorithm must translate chain codes to actual co-ordinates. The following procedure processes data in the CHAIN DICTIONARY

(a) from the chain header information get:

starting XY location,
 number of words containing chain codes for this
 feature,
 number of chain codes used in the last word;

(b) for each chain code word, unpack the chain codes;

(c) for each chain code, calculate the new XY
 co-ordinate:

```

for i = 2,1 do
  begin
    change(i)=0 ;
    if chaincode > 4 then change(i)=-1 ;
    if chaincode > 0 and chaincode < 4 then change(i)=1
  end ;
  x=x + change(1) * chainlength ;
  y=y + change(2) * chainlength ;
  
```

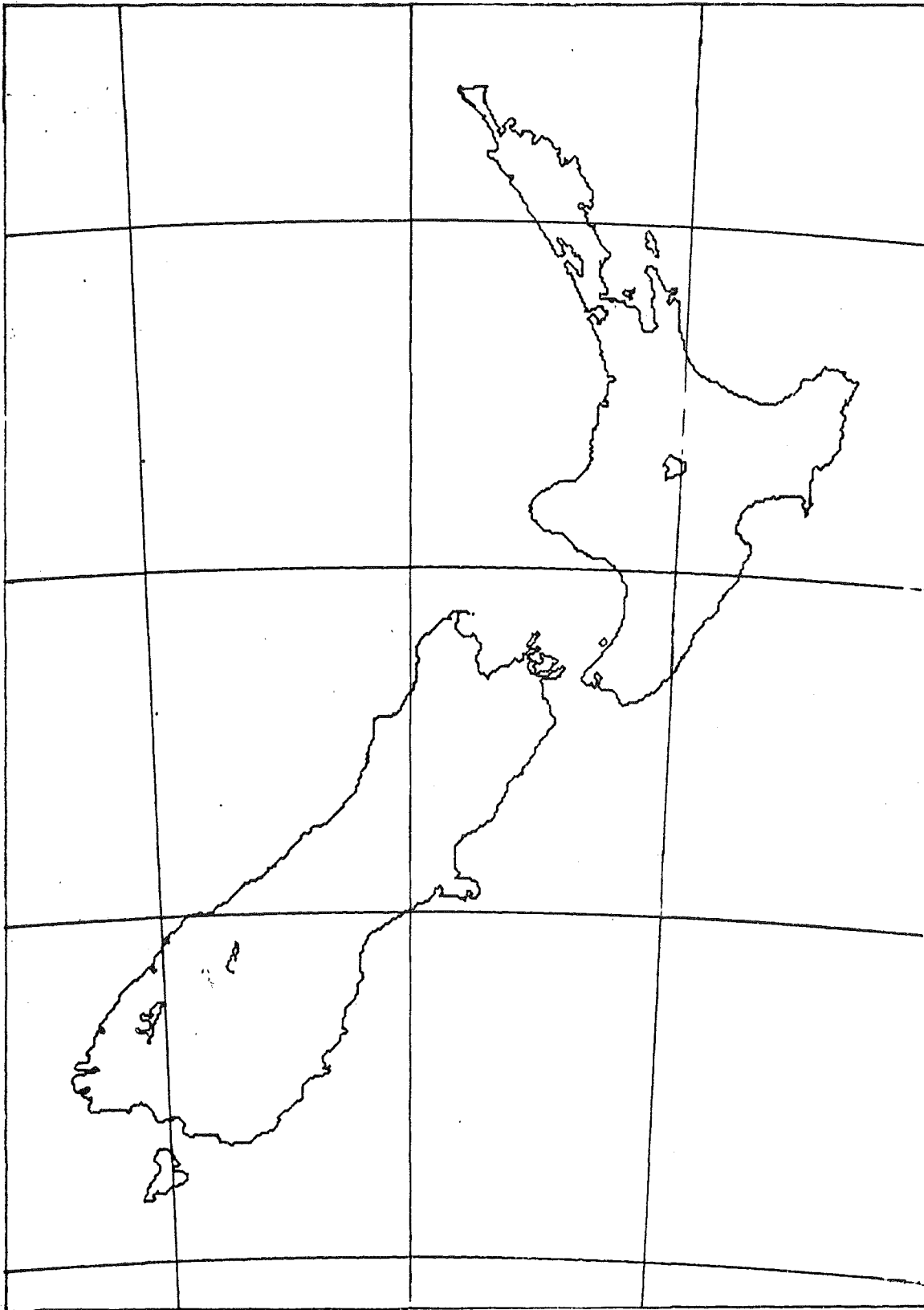


Figure 5-8a

Effects of reproducing chain encoded data

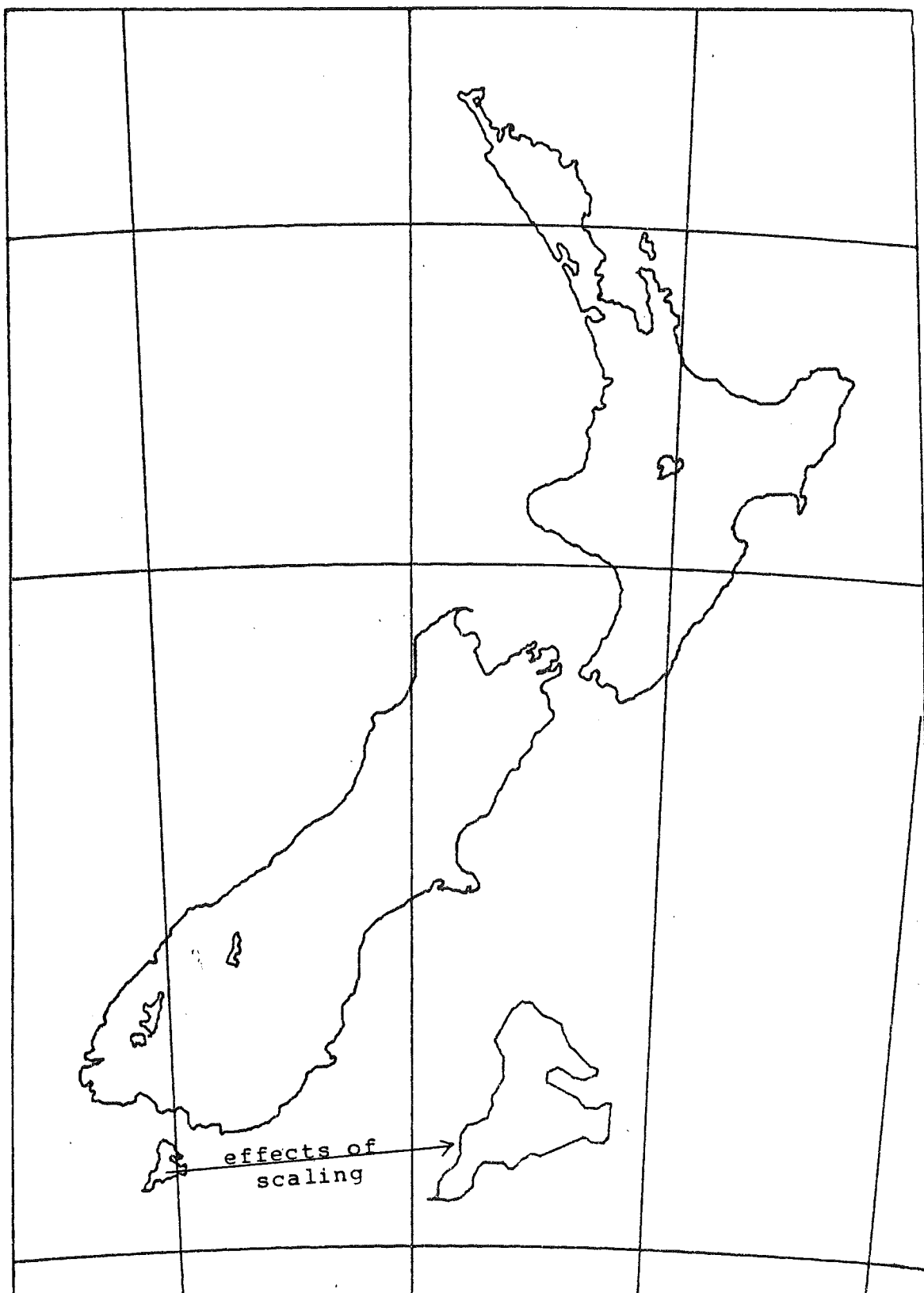


Figure 5-8b

Effects of reproducing chain encoded data
when smoothing algorithm applied

Using an increment size of .4mm, a 3-bit chain code gives a visual saw-tooth effect when applied (see Figure 5-7a). A saw-tooth effect is cartographically unacceptable, therefore three options were considered:

- (a) decrease the chain length,
- (b) use a 4 bit chain code,
- (c) modify the plotting algorithm.

A decrease in chain length resulted in a chain file larger than the initial raw data file, without correcting the saw-tooth effect. In general, it appears that a 3-bit chain code will never produce a visually smooth curve. The 4-bit chain code increased the file size due to the packing density, but removed the saw-tooth effect by giving a minimum change in direction of 22.5 degrees. However, a smoothing algorithm giving the packing density of a 3-bit chain code and the visual effect of a 4-bit code was implemented. The algorithm connects the mid points of successive 3-bit chain codes rather than the end points. This has the effect of providing a change in direction of 22.5 degrees, provides efficient packing, and permits an easier encoding/decoding algorithm. Figure 5-7b illustrates the effects of this smoothing algorithm, as well as its effects when enlarging the chain data.

5.3.4 Character Set

Quality of lettering is a factor that contributes substantially to the overall quality of the work. Two major considerations are the individual character formats and the placement of character strings. The majority of character sets supplied by manufacturers are restricted to characters approximated by straight line segments. This restriction is

cartographically inelegant, and so the author designed and implemented a character set with the aim of:

- (a) providing any user-defined character or symbol,
- (b) generating symbols using both straight lines and curved lines,
- (c) providing a variable displacement between adjacent characters in a string rather than limiting characters to a fixed size matrix,
- (d) providing character manipulation functions such as the rotation of text strings,
- (e) simplifying software portability.

Through the XY DICTIONARY coding scheme, it is possible to implement a character generator satisfying the above objectives. Characters are stored in a 2 dimensional $m \times n$ matrix and are generated from both straight line segments and circular arcs. The value of n for the matrix varies according to the properties of a particular character, thus providing a variable displacement factor between adjacent characters. Any symbol defined by the user in terms of lines and circular arcs can be incorporated.

The character generator uses a LOCATION TABLE and a SYMBOL DICTIONARY. The integer value of a character provides an index into the LOCATION TABLE which gives the corresponding start address of that character in the SYMBOL DICTIONARY. The format of the SYMBOL DICTIONARY is

X	Y
length x axis	no. of XY pairs
X ₁	Y ₁
X ₂	Y ₂
:	:
:	:
X _n	Y _n
length X axis	no. of XY pairs
X ₁	Y ₁
:	:

where the XY pairs use the coding scheme described in Chapter Four. If the first character in a string is plotted at location $(X_S Y_S)$, subsequent characters use the previous character's starting location modified by the current character's length of X axis value, to generate their starting location. This mechanism provides variable displacement of characters relative to the X axis and their predecessors. The number of XY pairs identifies the number of consecutive code words in the SYMBOL DICTIONARY used to define the symbol. The codes are processed by the XY DICTIONARY decoding algorithm.

Functions required for character string manipulation include scaling, rotation and slanting of symbols. Rotation allows text strings to be plotted at any angle to the X axis, and slanting gives an italics mode. To provide these facilities, a transformation matrix is used. The general form of the transformation matrix is given in Figure 5-11a, where $a, b, c,$ and d affect scale, slant and rotation, m and n are translation factors, p and q are projection factors, and s an overall scaling factor. Each character co-ordinate pair is formed into a 3 element vector $[x, y, 1]$ (Rogers 1976) and used by the transformation matrix routine.

As rotations occur about the origin, all points must first be transformed to the origin, rotated, then retransformed to their original location. The specific manipulation matrices for scaling, rotating, and transforming are shown in Figures 5-11b to 5-11e. Because matrix multiplication is non-commutative, the order in which transformations are applied is critical. Figure 5-11f gives a single matrix which is the product of $BxCxDxE$. It performs the functions of those matrices in a single operation.

The input to the character generator is the character

string, the position at which to begin plotting it, the scale, and the angle of rotation in degrees.

The italics mode facility is provided by manipulating the b and c factors in the general matrix. The b factor produces shear in the Y direction while the c produces shear in the X direction. To produce an italics effect in a string parallel to the X axis, a slant factor c is included in the operation.

A mechanism for providing differing line widths for character strings is implemented as a function of the data type. For instance, the plan numbers in the PLAN file are to be drawn with a thicker line width than individual parcel data such as section numbers and areas. If multiple pen heads exist, the plotting software can be programmed to selected the appropriate pen, otherwise the user must change pens manually when required.

The author implemented two character sets (Figure 5-9), one for the plotter and the other for the graphics display terminal. The latter set was designed for high speed displays and therefore approximated all symbols by straight line segments, thus removing the arc generation overhead. For characters requiring circular arcs, each circumference co-ordinate on the generated arc must be transformed if the angle of rotation is not zero, thus adding to the arc generation time.

For implementation of the character set, the alphabetical characters and the survey symbols were manually drafted and their co-ordinates and codes input by the digitising procedure.

ABCDEFGHIJKLM

OPQRSTUVWXYZ

01234567890

UP

LARGE OR SMALLER

STILL

ANY ANGLE

THROUGH 360

DEGREES
DOWN

EVEN ITALICS

NOTICE SPACING

BETWEEN LETTERS

Figure 5-9a

Interactive VDU character set

Sample Character Sets.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890 -+.,&

Italics are possible at any slope

This is at 15 degrees slope.

Taller characters are possible,

as are wider.

*Any angle thru 360 degrees
Many combinations are possible.*

Figure 5-9b

5.3.5 Visual Centres

The location of text is another important factor that contributes to the appearance of the final document. Two options are available for obtaining the location for plotting text strings:

- (a) predefining the location and storing this information in the data base,
- (b) calculating a centroid from co-ordinate data in the XY DICTIONARY.

In the data base, co-ordinate data is correlated with features at a parcel level through information in the POINTS file. Features that are not explicitly defined by their POINT data, such as PLANS that reference several parcels, do not have POINT data for generating centroids. This information can be obtained using the algorithm presented in Chapter Six. However, such a location may lie across an individual parcel's centroid.

For individual parcels, the author proposes generating centroids using data in the POINTS file and XY DICTIONARY. Due to the large number of individual features, it is expensive to hold centroid data for each feature. An algorithm for generating the visual centre of any irregular polygon was implemented. Such an algorithm must:

- (a) have a high percentage of successful first attempts at generating an acceptable centroid, and
- (b) provide additional techniques to cater for a first attempt that does not produce an acceptable centroid.

A geographical centroid (Waugh 1973) is not necessarily acceptable, as often this falls outside the feature's boundaries.

For features that do not have explicit POINT data, and features, such as roads, for which it is preferable to hold centroid data, the centroid data is digitised and held in the PLOT file. Six bytes of information per entry are:

- (a) 4 bytes for the XY starting location,
- (b) 2 bytes for a pair of XY co-ordinates relative to the starting location to give the angle of rotation.

The algorithm for generating an acceptable centroid for an irregular polygon is as follows:

- (a) Preprocess Co-ordinate Pairs. Co-ordinates associated with access ways to a feature are removed (see Figure 5-10). A procedure iteratively scans the feature's co-ordinates, removing consecutive co-ordinate pairs $C_i C_j$ in which:

- i) the length of the line segment between C_i and C_j , divided by the area of the feature, is less than a factor F ,
- ii) the sum of the two angles C_{i-1}, C_i, C_j and C_i, C_j, C_{j+1} is approximately equal to 180 degrees,

until no further reductions can be made. These two conditions are specifically designed to detect points defining narrow access ways to a feature.

In Figure 5-10, points 1 and 9 will be removed in the 1st scan, 2 and 8 in the 2nd, and in the 3rd points 3 and 7 are eligible by criterion (i) but rejected by criterion (ii). Consequently points 3 to 7 inclusive remain after this phase.

- (b) Generate Bounding Rectangle. A rectangle is constructed parallel to the base axes from both the

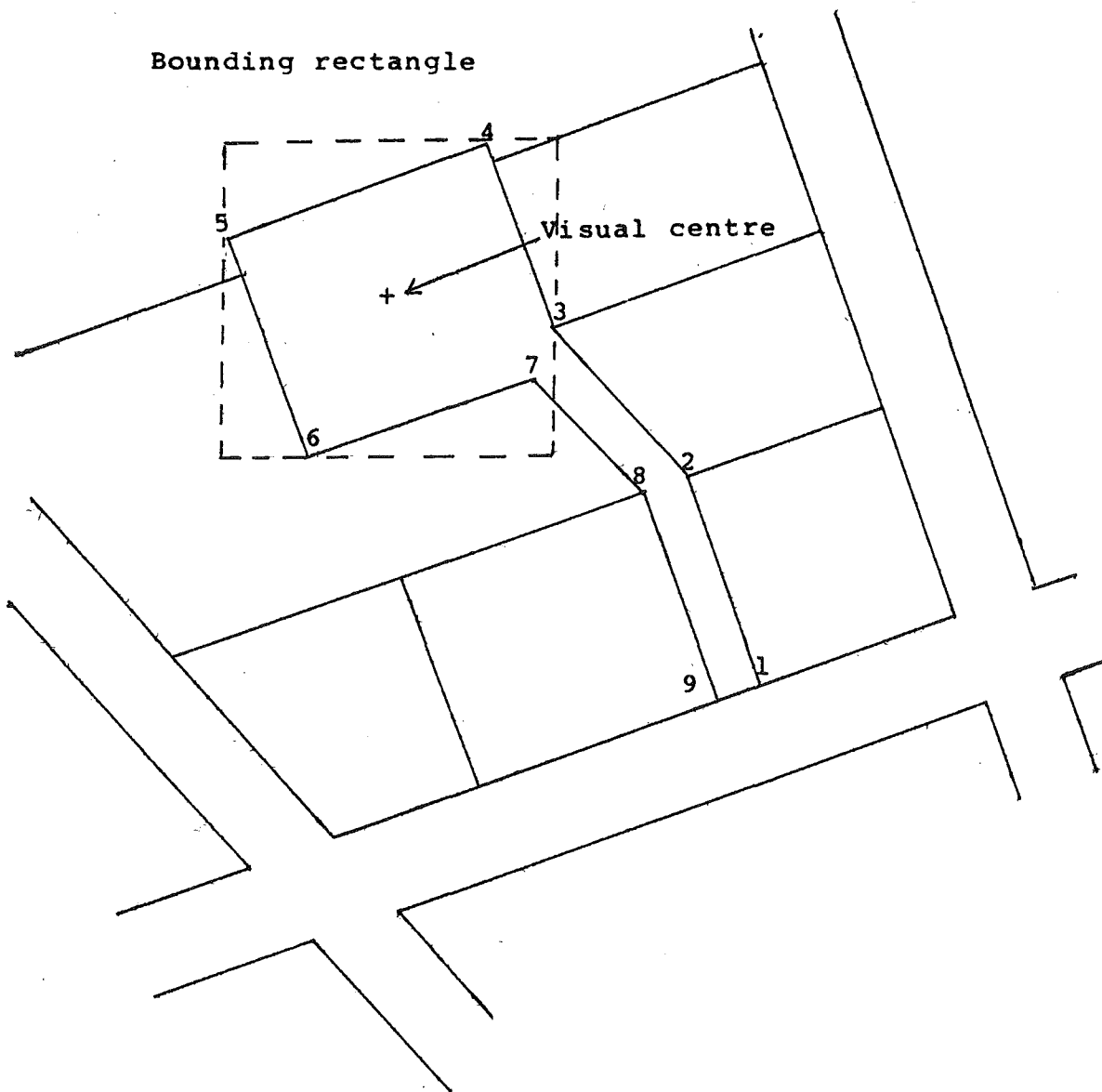


Figure 5-10

Example of Visual Centre calculation

minimum and maximum X and Y co-ordinate values passed from phase (a). The feature concerned lies within this rectangle, except for those access ways removed in the previous phase.

- (c) Calculate a Visual Centre. A visual centre for the feature is the midpoint of the rectangle generated in the previous phase.
- (d) Test for Suitability. To test the point for its suitability we must:

- i) determine whether the point is within the feature,
- ii) decide on the angle at which to plot the text data.

One technique for determining whether a point lies inside an irregular polygon is by counting the number of times a line drawn from the centroid, parallel to either the X or Y axis, crosses the polygon's boundaries. For an inside point, the count is odd, while for an outside, the count is even.

A test is then made to determine whether the character string will fit within the feature when plotted horizontal to the X axis. This provides optimum legibility. If this fails, an attempt is made to plot the data at the same angle as the longest side remaining after phase (a).

- (e) Alternative Procedures. If an unacceptable point is generated above, alternative techniques to cater for exceptional situations are required. One such technique is:

- i) get the midpoint of the longest side,

- ii) get the midpoint of the longest adjacent side,
- iii) connect these midpoints and calculate the midpoint of the resulting line.

This can then be used by phase (d).

Calculating visual centres for features of a sufficient area can bypass the above algorithm and generate a point from the midpoint of a line connecting the midpoints of the two longest sides. For small scale maps, such as 1:10000, often it is not possible to generate acceptable centroids for small features. In this instance, the descriptive data must be plotted outside the feature but connected by an arrow.

$$\begin{array}{ccc}
 \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & s \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} & \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)} \\
 \\
 \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} & \\
 \text{(d)} & \text{(e)} & \\
 \\
 \begin{bmatrix} a\cos\theta & a\sin\theta & 0 \\ -a\sin\theta & a\cos\theta & 0 \\ -am\cos\theta + a\sin\theta & -am\sin\theta - a\cos\theta & 1 \end{bmatrix} & & \\
 & \text{(f)} &
 \end{array}$$

Figure 5-11

Example of co-ordinate Transformation matrices

CHAPTER SIX

THEMATIC MAPPING

This chapter dicusses the extensions necessary to provide a thematic mapping capability for the cadastral system described in Chapter five. The cadastral data base provides a basis for cadastral mapping functions, but does not cater for thematic mapping requirements. Incorporating additional data sets and providing data manipulation facilities gives a data base that can be used as a multi purpose cadastre as well as a geographical processing system. The following sections introduce the basic techniques for providing these capabilities.

6.1 SMALL SCALE MAP PRODUCTION

This section describes a technique developed to produce small scale cartographic maps from the data base's large scale base maps. For example, four neighbouring 1:1000 scale base maps can be merged to form a 1:2000 scale map. The reduction and merging of base maps requires:

- (a) automatic access to all required base maps,
- (b) the merging of common boundary points on adjacent sheets that do not have identical boundary co-ordinate values.

Data base map names contain information about their location in a 1:50000 grid system covering the entire country. The 1:50000 grid system is further decomposed into smaller grid units depending on the scale of the base map. Map names contain:

- (a) a reference to the appropriate 1:50000 grid section, for example, R27,
- (b) the scale of the map, for example, 1000,
- (c) a number in an easterly row of the 1:50000 grid, for example, 34,
- (d) a number in a southerly column of the 1:50000 grid, for example, 12.

The data base's filename for the example is R2713412.ext, where ext identifies the file type. The XY DICTIONARY has XYD as its type whereas the CHAIN DICTIONARY has CHN for its type. Surrounding sheet names are calculated using the easterly and southerly identifiers, thus providing automatic access to the required files in the data base.

Any type of feature may occupy more than one base map, in which case it is referred to as a common boundary feature. In general, the common boundary point merging technique bypasses unequal boundary points by connecting their end points directly. Any map has four boundary faces B1, B2, B3, and B4 (Figure 6-1), where face pairs B1 and B3, and B2 and B4 represent common boundaries on neighbouring maps. Face B1 on map A shares a common boundary with face B3 on map B.

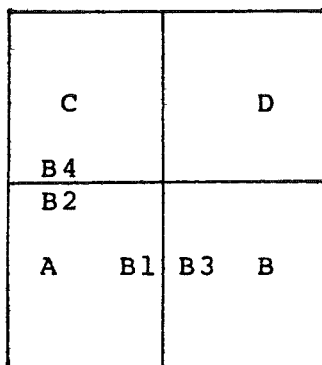


Figure 6-1

Example of 4 map sheets

Depending on the face, either the X values should be equal for neighbouring maps and the Y values 0 to MAXY respectively, or the Y values equal and the X values 0 to MAXX. However, as base maps are independently created, using different transformation matrices, common boundary points will seldom be equal. Hence, a merging procedure is required.

To produce small scale maps, a scale factor and the name of the base map located at the south west corner of the new map, provides a set $S = \{map_1, map_2, \dots, map_n\}$ of required map names. The technique for generating small scale maps is:

- (a) Get a map name from S,
- (b) For those boundary faces not yet processed as a result of sharing a common boundary with a previously processed map from S:

- i) get the XY DICTIONARY's for the two maps with a common boundary,
- ii) for each DICTIONARY, record in an ordered table the appropriate xy boundary co-ordinate pairs and their location in the XY DICTIONARY. For example, when processing B1 in Figure 6-1, co-ordinate pairs whose x value equals MAXX are stored in a table for map A in order of increasing y values, while those for B3 are also held in order of their y values, but for co-ordinates whose x value is 0.

- (c) Each table contains pointers to the XY DICTIONARY as well as to co-ordinates and their codes. If B_i is a boundary point from table i, and P_i is the point in the XY DICTIONARY connected to B_i , then the codes for $P_i \rightarrow B_i$ will define either a line

feature or an arc. If the code defines:

- i) a line, then a line is drawn from P_i to P_j , thus bypassing boundary points B_i and B_j which may not be equal. By maintaining ordered tables and processing them sequentially, we can be certain that B_i and B_j are intended to have the same boundary co-ordinates. Before drawing the line, co-ordinates P_i and P_j must be modified by the respective offsets of their base maps.
- ii) an arc, then processing depends on the nature of the arc. If the arc starts and ends on the same sheet, (Figure 6-2a), then an arc is constructed using the 3 co-ordinates C_1 , C_5 , and C_6 in Figure 6-2a. This provides a minimum deviation and bypasses the four boundary co-ordinates. If the arc starts on one sheet and ends on another, (Figure 6-2b), then boundary points B_i and B_j are averaged to give a new point B_a , and the arc is constructed using points C_1 , B_a , and C_6 in Figure 6-2b.

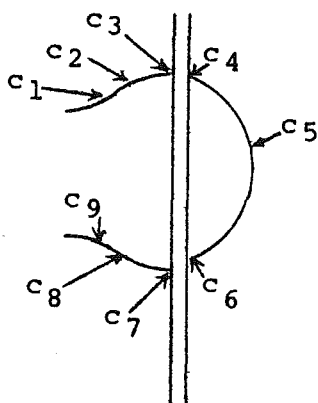


Figure 6-2a

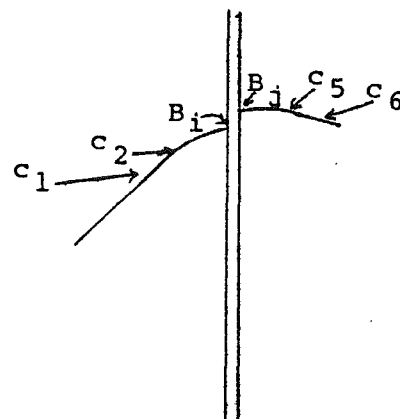


Figure 6-2b

Each map in S is initially processed to generate only

the boundary features for the new small scale map. Finally, each map in S is then processed to construct the features local to it, ignoring any boundary features that were processed in the first pass.

CHAIN DICTIONARY features that cross sheet boundaries represent potential problem areas. A technique catering for stream digitised features suppresses drawing n chains before the sheet boundary on each sheet, thus providing two co-ordinates B_a and B_b , and connecting these co-ordinates by a straight line (Figure 6-2c). This technique prevents the possibility of a sudden 90 degree change required to connect unequal boundary points directly.

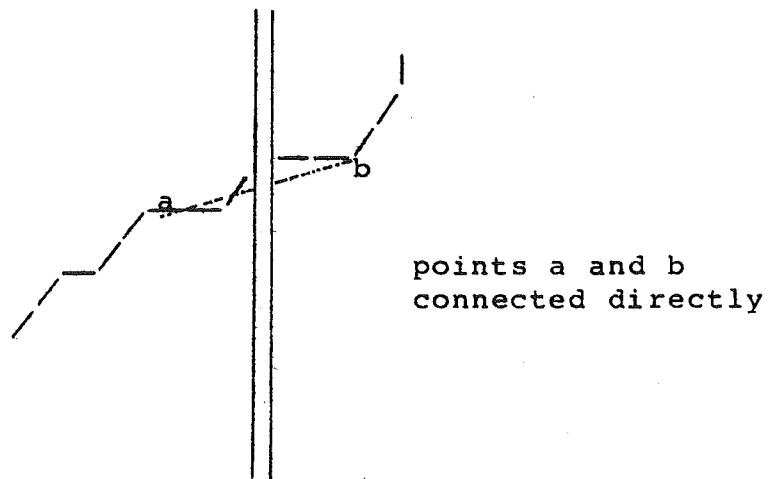


Figure 6-2c

Plotting of text uses the technique discussed in section 5.3, but reduces the basic scales for the text data. For very small scale maps, it will not be possible to provide all the annotations required. In this case, a subset such as plan types and numbers may be produced.

6.2 ADDITIONAL DATA SETS

The data base structure (Figure 4-5) is designed to simplify the inclusion of additional data sets. New data sets can be both parcel and non parcel oriented sets catering for

the 1:1 and 1:many relationships described in Chapter four. Non parcel oriented data will involve thematic and geographical data. Parcel oriented data can be included in the appropriate level of the hierarchy emanating from the PLAN TYPE DIRECTORY, while non parcel oriented data provides new feature files in the data base. All additional data sets are held in separate files. Examples of additional data sets and methods for including them are:

- (a) valuation data: will be held on a parcel basis, and therefore included in the hierarchy at the same level as section/lot numbers.
- (b) parcel addresses: will be held on a section/lot number basis and comprise a street number and a link to the NAME file for ROADS/RAILWAYS.
- (c) census tracts: this will require digitising census block boundaries and storing them in a new structure called a CENSUS DICTIONARY if the points in the XY DICTIONARY are not sufficient for their definition. A new feature type will be defined for census blocks which contains:
 - i) census block information data,
 - ii) POINTS data which references the co-ordinates in the CENSUS DICTIONARY that define the census block.

Census data will be digitised after modifying the digitising process to accept a CENSUS menu command, and it may use the XY DICTIONARY coding structure.

- (d) soil, geological, and vegetation overlays will all be held as discrete polygons using the XY DICTIONARY coding scheme. Each item is to constitute a new feature in the data base.

6.3 DATA MANIPULATION SUBSYSTEM

Using the data structures described in this thesis, a land information system containing cartographic and geographical data is possible. The system must be capable of processing data to extract interrelationships between different data sets and to produce:

- (a) special purpose cadastral maps,
- (b) statistical summaries and thematic maps displaying geographical data relationships.

6.3.1 Feature Display

The POINT files associated with particular features provide the means for displaying individual features. POINT file data for a given feature references both XY and CHAIN DICTIONARY data that generates the feature. When generating individual features from POINT file data, the XY DICTIONARY codes apply for those features that reference them.

If POINT data for a particular feature type occurs at level n of the data structure hierarchy (Figure 4-5), features at levels 1 through to n of the hierarchy can be displayed without special manipulation algorithms. For example, a specific parcel, all parcels of a specific plan number, or all parcels of a specified plan type can be displayed. Similarly, a base map displaying only streets and/or census block boundaries can be constructed, and using the small scale map production technique, a number of maps may be combined to produce the required display.

6.3.2 Feature Aggregation

Production of a special map, such as a map of plan boundaries, requires a technique that extracts internal parcel boundaries and plots only the external plan boundaries. A

special algorithm is required because POINT data is held on a parcel basis and not a plan basis. An algorithm developed by the author generates equivalent POINT data for plans from the POINT data of the parcels local to the plan. The algorithm which uses the POINT data of the features in the entity is:

- (a) Standardise POINT identifiers for all features. Any xy co-ordinate values that are the same, but referenced by different POINT identifiers (for example, points 54 and 112 in the corrected version of Figure 5-7), must be referenced as either 54 or 112.
- (b) Generate a set S of point numbers whose co-ordinate values represent either the maximum or minimum X and Y values of the features concerned. S will contain either 3 or 4 boundary points.
- (c) Add to S any points referenced only once by all the features, and the points connected to either side of them, as these points must also be boundary co-ordinates.
- (d) Let C be the first point in S.
- (e) Using POINT data for all features, determine all possible paths from the current point C.
- (f) If a path from C leads to a point P which is in S then
 - if all points in S have been processed and P is the point obtained in (d), then the external boundary completes at P and processing terminates
 - else
 - if P has not been processed, then P is a point on the external boundary path. Remove P from S, let C equal P and return to (e)
 - else
 - the path to P represents a closed polygon that defines an internal feature's boundary. Therefore, this route is rejected as a possible boundary route
 - else
 - if another path exists from phase (e) that hasnt

```

        been processed then get the P value and
        return to (f)
    else
        for each P generated in (e), let C equal P
        and continue processing at (e)

```

This algorithm will generate the external boundary points of the required entity. Appendix Five demonstrates the use of this algorithm. It is useful for producing special purpose maps and also for calculating visual centres for non parcel oriented features.

6.3.3 Query Language

An interface between the user and the data base can be implemented through a query language which provides access to:

- (a) the analytical procedures used to generate special purpose displays,
- (b) the procedures that traverse the data structures to process information.

The language will be similar to the interpretive languages discussed previously, in that it will contain predefined COMMANDS with optional parameters.

Before query language commands can be processed, all data relationships must be constructed, as these are not explicitly held in the external files. Data relationships are established when the external files are used to generate internal files. For example, the 1:many relationship between plan numbers and their corresponding parcels is generated by creating a parallel table of pointers for each plan number. As the first parcel for each plan is put into its internal file, its offset within the file is held in a plan number pointer file. Pointer files need only be created for 1:many relationships. 1:1 relationships are implied by the same offset into parallel tables.

An interactive query system must provide the user with the ability to interrogate the data base by specifying items of interest on a display screen. For each feature with POINT data, a visual centre point is generated and displayed in the feature. To identify the feature of interest, a light pen or cross hair cursor is positioned at this point. The software system has a table of these xy centre co-ordinates, and for each co-ordinate pair, a pointer to the appropriate feature is held (see REFERENCE POINTS in Figure 4-4). Details of the feature can then be obtained. For example, suppose a parcel's plan and lot numbers are unknown, but the parcel can be located visually on a map, then this mechanism provides the ability to access the required units data.

CHAPTER SEVEN

CONCLUSION

The following observations have been made after using the experimental MAPPAK system extensively for both rural and urban cadastral map compilation.

Firstly, the success of an interactive system can be measured by its ease of use and the time required to process a map completely through the system. For less than \$100,000, the hardware configuration described in section 2.3 offering interactive facilities was purchased. Both the hardware and experimental software system developed provided an economic approach to cadastral map compilation, storage, and display. Few errors were encountered during digitising that required any editing prior to producing a hardcopy plot of the document. Those that did occur could easily be rectified using an interactive editing procedure.

The times required to digitise maps varied depending on the scale and the amount of detail. A densely covered 1:2000 base map containing extensive hydrographic features took approximately 2 hours to digitise completely. The same sheet if processed by current automated scanning techniques would require time to

- (a) separate the features prior to scanning,
- (b) convert the raster data to vector format, and
- (c) merge the independently scanned features to generate a GDS.

Consequently, interactive digitising is the most economical means for converting cadastral maps from the old imperial mapping grid to the new mapping grid and simultaneously

generating a digital data base for cadastral maps.

The second observation was that generation of the DDS for a map suffers from the following problems:

(a) inclusion in the data base. Descriptive data is entered manually after generation of GDS. The large amounts of data require considerable manual processing. For example, establishing POINTS data files for a map's features requires correlating each feature with its individual co-ordinates. This process is done either interactively or by extracting co-ordinate identifiers from a proof plot. This process is both time consuming and error prone.

A better method might be the use of audio input. Neroth (1974) and Beck (1977) describe graphics systems using audio input systems capable of recognising 60 keywords. The audio input subsystem provides a lower error rate on input than conventional methods, and gives both visual and audio output facilities. An interactive digitising system with an audio input facility would provide a means of capturing descriptive data during digitising. This is not done currently as it involves alternating between the digitiser and a keyboard, which is both time consuming and inefficient in resource utilisation. A modified interactive system would allow the operator to enter parcel details orally prior to digitising the feature, and on digitising the co-ordinates, the POINTS file for the feature would be generated automatically.

In general, the descriptive data files could be constructed simultaneously with the GDS with little additional time being required.

(b) storage requirements. In general, most articles on cartographic data bases emphasize the importance of organising

the GDS to reduce its storage requirements, without referring to the organisation of a DDS. The XY DICTIONARY is effective in this respect, and the CHAIN DICTIONARY can compress hydrographic data files to less than one-fifth their original size. However, the GDS files required only a small percentage of the total file space needed for storing a cadastral map. The techniques used to compress GDS data, although not applicable on large systems, were found to be necessary for this system which has limited on-line storage.

(c) plotting times. The plotting time for GDS data is negligible compared to that required for annotating the map with its descriptive data. For an average map, plotting times for GDS data vary between 7 and 13 minutes depending on the amount of hydrographic data present. The addition of DDS data can take another 2 to 3 hours. Under such circumstances, a plotting algorithm that plots features in the same order as they were digitised, without attempting to optimise the plotting sequence, is justified.

In general therefore, the processing of DDS data proves to be the greatest single factor in restricting the throughput of the system, and attempts to improve this aspect to map making are required.

The final observation concerns the chain encoding technique. To obtain a cartographic reproduction of hydrographic features, a small chain length is required. As a result of the chain length, it was noticed that for a 3-bit chain code, a change in direction is always plus or minus 45 degrees. This implies that a sequence of 3-bit chain codes will differ by only the least significant bit for successive codes. Depending on the nature of the data, this observation may also apply to 4-bit chain codes. Consequently, it is

possible to further reduce the storage size of chain encoded data files using a single bit to indicate the change in direction. The following system performs this compression:

- (a) record the starting co-ordinate and the initial direction of movement for a chain feature.
- (b) encode the data as follows. A series of 'n' similar bits indicates the number of chain lengths to move in the current direction. A change in direction is indicated by a change in the bit pattern. The direction of change is indicated by the bit following the change in direction bit. A 0 bit indicates a change in direction of plus 'm' degrees and a 1 bit indicates a change of minus 'm' degrees from the current direction. The change of direction bit is repeated 'n' times for the number of chain lengths to proceed in the new direction.

This technique reduces a chain encoded data file by at least a third when no two successive chain codes are the same. However, chain-codes generally tend to occur in clusters of similar codes, and therefore considerable savings can be achieved using this technique.

Appendix Six contains a sample of a 1:10000 scale base map generated from four imperial record sheets using the experimental software system. The map shows that the techniques discussed in this thesis are satisfactory for the purpose of cadastral map compilation.

ACKNOWLEDGEMENTS

Firstly, I wish to express my appreciation to both of my supervisors; Professor J. P. Penny for his valuable guidance in the preparation of this thesis, and Doctor Zen Loy for his assistance with the technical aspects of the software.

I would also like to acknowledge the New Zealand Department of Lands and Survey who provided the special hardware necessary to make this project possible, and in particular, Graeme Crocker for both his help regarding the New Zealand cadastral map requirements and his assistance in the preparation of diagrams. Also, I wish to thank Mike Bundock whose programs provided the hardcopy plotter documents.

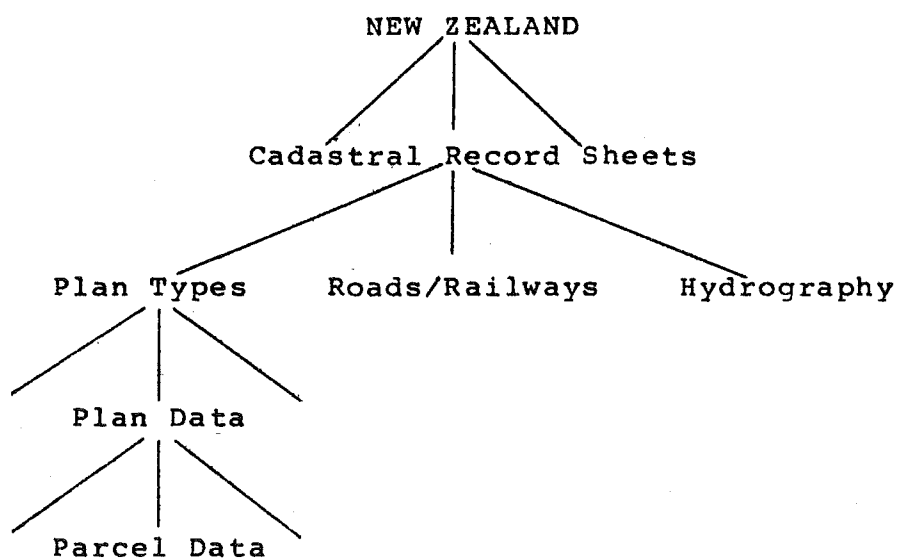
REFERENCES

- ARMS, A. (1967) MAP/MODEL System: System Description and Users Guide, Bureau of Governmental Research and Service, University of Oregon. 59p.
- ATKEY, R.G, GIBSON, R.J. (1975) Progress in Automated Cartography. Paper presented at Commonwealth Survey Officers Conference. 13p.
- BAXTER, R.S. (1976) Computer and Statistical Techniques for Planners. 1st ed. Methuen & Co. Ltd. 336p.
- BOYLE, A.R. (March 1978) Scan to Line Vector Conversion: Program and Method Description. Paper from National Research Institute for Mathematical Sciences, NRIMS/W/75/9. Internal Report I 115, Pretoria. 27p.
- BEEK, B. (1977) Voice Data Entry for Cartographic Applications. Proceedings of the American Congress on Surveying and Mapping, October 18-21, 1977. 161-185.
- CLOTS (1978) Clots User Manual. Department of Lands, Adelaide, South Australia.
- COOKE, D.F, MAXFIELD, W.T. (1967) The Development of a Geographic Base File and its use for Mapping. In Urban and Regional Information for Social Programs (1975). 207-218.
- FREEMAN (1961) On the Encoding of Arbitrary Geometric Configurations. Transactions of the IRE, Vol EC-10, No. 2. 260-268.
- GORRINGE, H.P, BOYLE, A.R (March 1978) An Attempt to Simplify Digitisation in Automated Cartography. Paper from National Institute for Mathematical Sciences NRIMS/W/75/9 Note 41, Pretoria. 18p.
- KROLL, F.S, BOYLE, A.R (1974) Online Collection and Manipulation of Two and Three Dimensional Topographic Data with a New System at HANSA LUFTBILD. Paper presented at 7th International Conference in Cartography, Madrid, April 28th - May 4th. 31p.
- LARSSON, G. (1975) Social and Administrative Functions of a Cadastre: Environmental Studies, Planning and Administration. The Canadian Surveyor Vol 29, No. 1, March 1975. 75-81
- MONMONIER, M.S (1970) The Scope of Computer Mapping. From Bulletin of Geography and Map Division No. 81. American Geographical Society. 2-13.
- MOYER, D.F, FISHER, K.P. (1973) Land and Parcel Identifiers for Information Systems. 1st ed. American Bar Foundation. 345p.
- NEROTH, C.P. (1975) A Graphical Programming System with Speech Input. Computer Graphics Vol. 1, 1975. 227-231.

- PEUCKER, T.K (1973) The Interactive Map in Urban Research, Final Report after Year One, University of BC.
- PEUCKER, T.K, CHRISMAN, N. (1975) Cartographic Data Structures. In The American Cartographer Vol. 2, No. 1. 55-69.
- ROGERS, D.F, ADAMS, J.A (1976) Mathematical Elements for Computer Graphics. 1st ed. McGraw-Hill. 239p.
- SCHMIDT, A.H. (1969) The Automap System. Surveying and Mapping Vol. 29, No. 1 (March). 101-106.
- SCHMIDT, A.H, and others (1974) Cartographic Data Structures Panel. In American Congress on Surveying and Mapping (1977). 165-181.
- STREET, G.S.B, WOOD, P.A. (1975) The HRD-1 Laser Display System. Paper presented to Technical Working Session, April 21st-25th. In Automation in Cartography Commission III. 184-192.
- SWITZER, W.A. (1975) Canada Geographic Information System. In Automation in Cartography Commission III. 58-81.
- TAYLOR, D.R.F. (1973) The Canadian Cartographer and the Computer: Present Trends and Future Challenges. Cartographic Monograph No. 9. 1-9.
- TOMLINSON, R.F. (1973) A Technical Description of the Canada Geographic Information System, unpublished manuscript 14p.
- TOMLINSON, R.F and others (1976) Canada Geographic Information System. In Computer Handling of Geographical Data. UNESCO Press 27-73.
- TOMLINSON, R.F and others (1976) The Polygon Information Overlay System. In Computer Handling of Geographical Data. UNESCO Press 74-89.
- TURKE, K. (1978) Digitising of Geometric Data for Thematic Mapping. In Nachrichten aus dem Karten und Vermessungswesen , No. 35. Series II Translations. 85-91.
- WAUGH, W.C. (1977) GIMMS Reference Manual. Inter University/ Research Council Services. University of Edinburgh. Report No. 30. 95p.
- WEISS, C.C (1977) Evaluation of the GIMMS Computer Mapping Program. Census Field Study Cartography Series, No. 1. Statistics Canada. 80p.
- WILLIAMS, R. (1971) A Survey of Data Structures for Computer Graphics Systems. In Computing Surveys Vol. 3, No. 1. 1-21.

Appendix One

This appendix contains a description of the data set appropriate for the NZLS data base and, where appropriate, describes the file formats and the codes used. The data base contains both Graphical Data Set files and Descriptive Data Set files for cadastral record sheets, and is organised in a hierarchical nature as indicated below



The description of the data base is presented on a feature basis.

PLANS

This feature type has three hierarchical levels. Data elements in files at the same level of the hierarchy have a 1:1 correspondence with each other, while those at a lower level have a many:1 correspondence with higher level elements. In general, the T (toggle) bit in certain files permits the generation of the 1:many relationships. There are 5 major plan types:

- 1) Deposited Plans (DP)
- 2) Survey Office Plans (SO)
- 3) Maori Lands (ML)

4) Deeds

5) A Plans

Individual plans within each plan type have unique numbers and one or more lots or sections associated with them. Each lot/section in a plan has a unique number.

Plan Data. Data pertaining to individual plans is:

(a) PLAN NUMBER: the information header block in the XY DICTIONARY contains indexes into the PLAN NUMBER file that point to the first plan number for each plan type. In this file, all plans of a specific type are held on a contiguous basis.

(b) PLOT DATA: for each plan, the PLOT DATA gives the location at which to plot the plan type and number on the map.

Format:	0	15
	X ₁	
	Y ₁	
	X _r	Y _r

Description: X₁ - x value
 Y₁ - y value
 X_r - an offset relative to X₁
 Y_r - " " " " Y₁

(c) PARCEL POINTERS: this information does not exist in the data base. A table of pointers to the first lot/sec for each plan number is constructed from the T bit in the LOT/SEC file elements.

Parcel Data. The PARCEL POINTERS provide the 1:many relationship between items at the PLAN level of the hierarchy and the items at this level. Data files are:

(a) LOT/SEC: for other than Maori Land plan types, lot/sec details are:

Format:	0	1	2	15
	T	P	G	lot/sec #

Description: T - toggle bit. If set, then this sec-

tion is the first for the current plan number.

P - part indicator. If set, then this is a part section, for example, PART SECTION 47 DP 21745.

G - gazette bit. If set, then this parcel has a gazette reference. The order in which G bits appear in this file is the same as the order of the gazette details in the GAZETTE file. Consequently, there is not a 1:1 correspondence between parcel files and gazette files.

Maori land section identifiers are alphanumeric. Additional alphanumeric characters are appended to existing section identifiers to create new identifiers whenever a section is subdivided. Section identifiers are therefore variable in length.

Format:

0		15
T	Title ref	

Description: Title ref - a value indicating the 'nth' element of the TITLE file where the section number is stored.

- (b) AREA: an area measured in hectares containing 4 decimal digits is held for every lot/sec.
- (c) C/T REFERENCE: 6 bytes of alphanumeric data provide a certificate of title reference for every parcel.
- (d) POINTS: every feature contains references to the co-ordinates that generate it. There is a 1:many correspondence between a feature and the points that define it.

	0	12	15
Format:	T	Pointer	len

Description: T - toggle bit. Permits the generation of a pointer table similar to the PARCEL POINTER, that maps a feature to 'n' entries in the POINTS file.

Pointer - the XY DICTIONARY index of the appropriate co-ordinate or, an index to the CHAIN DICTIONARY.

len - if non zero, it specifies the number of consecutive XY DICTIONARY elements defining this feature, otherwise it indicates that the Pointer is to the CHAIN DICTIONARY.

(e) APPELLATIONS: a total of 70 appellations are used for land parcels. Some of the more common appellations used are:

1) Deposited Plans - Lot 7 DP 10248 being part section Block IX, Aria Survey District

2) Sections of

a) Survey Districts - Sect 14, Block VII, Kaeo SD.

b) Settlements - Sect 45, Puketi Settlement.

c) Villages - Sect 15, Village of Mamari.

3) Allotments of Villages - Allot 64, Sect 2, small lots near village of Panmure.

4) Maori Blocks - Sect 2B1, Anamahanga Maori Block, situated in Block VIII, Gore SD.

It is not possible to cater for all the appellations

used and a one plan system is a requirement before an efficient data base can be generated. The following system caters for 16 different appellations.

Format:

0	7	11	15
Title ptr	Type	Blck	

Description: Type - indicates the appellation type:

- 0 : Deposited plan
- 1 : Section of Survey District
- 2 : Section of Settlement
- ⋮

Blck - identifies the block number.

- (f) GAZETTE : the gazette details contain the year and page number of the gazette entry, as well as a brief description of the purpose of the land. This file is in the same order as the gazette bits in the LOT/SEC file.

Format:

0	7	20	31
Yr	Page	Title ref	

- (g) TITLE: this file contains all variable length alphanumeric data in packed character format. It contains Maori Land section identifiers, appellations, and gazette descriptions. References are to the 'nth' element in this file, rather than by a direct index into the file.

Format:

0	8	
0	# chars	1st reference element
1	C ₁	
2	C ₂	
	⋮	
	⋮	
n	C _n	
n+1	# chars	2nd reference element
	C ₁	
	⋮	

ROADS/RAILWAYS and HYDROGRAPHICAL features

For each of the features in this category, the following files are used:

- (a) PLOT DATA: contains the location at which to plot the name data for the feature. Format and descriptions are the same as for a plan's plot data.
- (b) POINTS: same applies as for a parcel's points.
- (c) NAME/TYPE: contains 19 bytes for the feature's name and a byte denoting the feature type, for example, Avenue, Road, Street or Lake, River, Harbour.

Appendix Two

This appendix contains a list of the items contained in the Information Header Block(IHB) of the XY DICTIONARY. The first record in the XY DICTIONARY is reserved for the IHB and room exists in the record for further items to be added as required.

GRID data values

Imperial Northing - 4 bytes
 Imperial Easting - 4 bytes
 NZMG Northing - 4 bytes
 NZMG Easting - 4 bytes

XY DICTIONARY

Number of xy co-ordinates - 2 bytes
 Head of free space thread - 2 bytes

CHAIN DICTIONARY

Number of chain encoded features - 2 bytes
 Chain length - 2 bytes

PLAN FILE data

Number of plans - 2 bytes
 Index to start of DP's - 2 bytes
 Index to start of SO's - 2 bytes
 Index to start of ML's - 2 bytes
 Index to start of DEEDS - 2 bytes
 Index to start of A Plans - 2 bytes

Number of parcels in PARCEL file - 2 bytes
 Number of gazette items in GAZETTE file - 2 bytes
 Number of ROADS/RAILWAYS features - 2 bytes
 Number of HYDROGRAPHICAL features - 2 bytes

BLOCK data

Number of Blocks - 2 bytes
 Block information
 Start of block definition in XY DICTIONARY - 2 bytes
 End of block definition in XY DICTIONARY - 2 bytes
 Block name - 20 bytes

Appendix Three

This appendix gives two FORTRAN algorithms for AFFINE transformation routines. The subroutine AFFINE sets up the initial transformation matrix. Subroutine TRANS transforms a digitiser co-ordinate to the new co-ordinate system using the affine transformation matrix generated by AFFINE.

```

SUBROUTINE AFFINE(DIGIT,PLOT)
C*
C*   DIGIT contains digitiser co-ordinates on input
C*   PLOT contains the required transformation co-ordinates
C*   Matrices DIGIT and PLOT are 3D arrays of the form
C*   A(i,j,k) where
C*       i identifies the appropriate map segment (1-4)
C*       j refers to x value (j=1) or y value (j=2)
C*       k identifies the map sheet corner (1-4)
C*
C*   An affine transformation matrix A is produced
C*   containing up to 4 affine transformations depending
C*   on the number of sheets being used to generate a
C*   single base map.
C*
COMMON/A/A(4,3,3)
REAL DIGIT(4,2,4),PLOT(4,2,4),B(3,3),C(3,3),D(6,3)
DO 100 I1=1,4
  IF(DIGIT(I1,1,1).EQ.0) GO TO 100
  I=1
  DO 1 K=1,3
    C(K,1) = DIGIT(I1,1,K)
    C(K,2) = DIGIT(I1,2,K)
    C(K,3) = 1.0
1  CONTINUE
  GO TO 8
C
2  I=2
  DO 4 K=1,3
    DO 3 L=1,3
      B(K,L) = D(K+3,L)
3  CONTINUE
  C(K,1) = PLOT(I1,1,K)
  C(K,2) = PLOT(I1,2,K)
4  CONTINUE
  GO TO 8
C
5  DO 6 K=1,3
    E = DIGIT(I1,1,4)*B(1,K)+DIGIT(I1,2,4)*B(2,K)+B(3,K)
    IF(E.EQ.0) STOP ' ERROR '
    F = 0.5*(PLOT(I1,1,4)*D(4,K)+PLOT(I1,2,4)*D(5,K)+
1    D(6,K))/E
    DO 6 L= 1,6
      C(K,L) = C(K,L)*F
6  CONTINUE
  DO 7 K=1,3
    DO 7 L=1,3

```



```

          A(I1,L,K) = B(K,1)*C(1,L)+B(K,2)*C(2,L)+B(K,3)*
1          C(3,L)
7          CONTINUE
          GO TO 100
C
8          DO 10 L=1,3
            DO 9 K=1,3
              D(L,K) = C(L,K)
              D(L+3,K) = 0.0
9          CONTINUE
            D(L+3,L) = 1.0
10         CONTINUE
            DO 16 J=1,3
              E = ABS(D(J,J))
              N = J+1
              IF(J.EQ.3) GO TO 13
              M = J
              DO 11 K=N,3
                F = ABS(D(J,K))
                IF(F.LE.E) GO TO 11
                E = F
              M = K
11         CONTINUE
              IF(M.EQ.J) GO TO 13
              DO 12 L=J,6
                E = D(L,M)
                D(L,M) = D(L,J)
                D(L,J) = E
12        CONTINUE
13         E = 1.0/D(J,J)
              D(J,J) = 1.0
              DO 14 L=N,6
                D(L,J) = D(L,J)*E
14        CONTINUE
              DO 16 K=1,3
                IF(K.EQ.J) GO TO 16
                E = D(J,K)
                DO 16 L=1,6
                  D(L,K) = D(L,K)-E*D(L,J)
16        CONTINUE
              GO TO (2,5) J
100        CONTINUE
          RETURN
          END

```

SUBROUTINE TRANS(RX, IX, RY, IY)

```

C*
C*   RX and RY are digitiser co-ordinates on input
C*   IX and IY are transformed co-ordinates on output
C*   MAP identifies the map segment associated with the input
C*   co-ordinates
C*   Matrix A is the affine transformation matrix
COMMON/A/A(4,3,3)
DOUDLE PRECISION Z(3)
C
DO 10 J=1,3
10  Z(J) = RX*A(MAP,J,1)+RY*A(MAP,J,2)+A(MAPJ,3)
    IX = Z(1)/Z(3)+0.5
    IY = Z(2)/Z(3)+0.5
    RETURN
    END

```

Appendix Four

The following FORTRAN subroutine is used to convert a series of xy co-ordinate pairs into a packed chain encoded format.

```

SUBROUTINE CHNCVT(CORD,CHLEN,CHAINS,NPOINTS,NTHCHN,
1          JTHCHN)
C*
C* This subroutine converts raw XY co-ordinates to chain
C* codes of unit length CHLEN in 1 of 8 possible
C* directions.
C*
C* CORD(i,1) contains raw x co-ordinates on input
C* CORD(i,2) contains raw y co-ordinates on input
C* CHLEN defines the chain length required on input
C* CHAINS() contains packed chain codes (5/word) on output
C* NPOINTS defines the number of input co-ordinates
C* NTHCHN identifies the last word used in CHAINS on output
C* JTHCHN indicates the number of chain codes packed into
C* the last word of CHAINS
C*
INTEGER CORD(1,1),CHAINS(1),CHLEN
INTEGER LAST(2),THIS(2),DIFF(2),CHXY(2),VAL,CHPTR,TRV
INTEGER OTH,ADJ
REAL MIDPT
DATA X,Y/1,2/
C*
C* INITIALISATION OF VARAIABLES
C*
MIDPT = CHLEN/2
NTHCHN = 1
JTHCHN = 1
CHAINS(1) = 0
LAST(X) = CORD(1,X)
LAST(Y) = CORD(1,Y)
C
DO 100 CHPTR=2,NPOINTS
THIS(X) = CORD(CHPTR,X)
THIS(Y) = CORD(CHPTR,Y)
POINT = 0
LASTPT = 0
TRV = X
OTH = Y
DIFF(X) = THIS(X)-LAST(X)
DIFF(Y) = THIS(Y)-LAST(Y)
IF(IABS(DIFF(Y)).LT.IABS(DIFF(X))) GO TO 10
IF(IABS(DIFF(Y)).EQ.0) GO TO 100
TRV = Y
OTH = X
C*
C* CALCULATE NUMBER OF CHAINS OF LENGTH CHLEN BETWEEN
C* POINTS
10 NPTS = DIFF(TRV)*1.0/CHLEN+(DIFF(TRV)/IABS(DIFF(TRV)))*
1      1.0*CHLEN)
IF(NPTS.EQ.0) GO TO 100

```

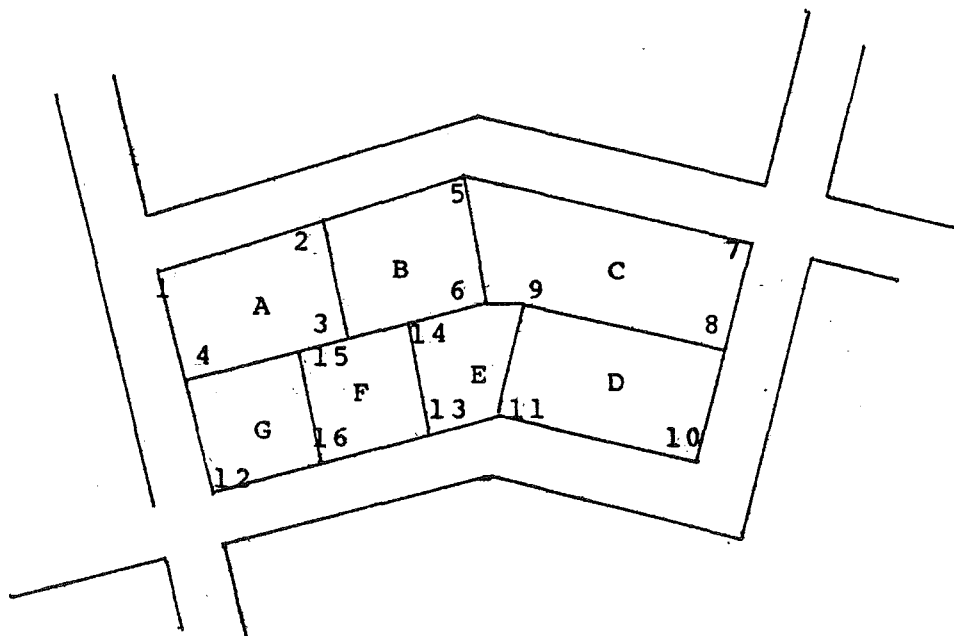
```

C*
C*   CALCULATE INCREMENT AND ADJUSTMENT
C*
      ANG = ATAN2(FLOAT(DIFF(Y)),FLOAT(DIFF(X)))
      IF(ANG.LT.0) ANG = ANG+6.28318531
      DEG = ANG*57.29577951
C*
C*   INCREMENT IS ALWAYS EVEN i.e 0,2,4,6
C*   ADJUSTMENT IS +1 OR -1 DEPENDING ON ANGLE
C*
      ADJ = -1
      IF(MOD(DEG/45,2).EQ.0) ADJ = 1
      INCR = MOD(DEG+45,360)/45
      FACT = ABS(DIFF(OTH)*1.0/DIFF(TRV)*CHLEN)
C*
C*   NEWPOINT = FACT * (X/Y) + C
C*
      NPTS = IABS(NPTS)
      DO 50 J=1,NPTS
        POINT = POINT+FACT
        IF(LASTPT+MIDPT.GT.POINT) GO TO 20
        VAL = INCR+ADJ
        LASTPT = LASPT+CHLEN
        IF(VAL.LT.0) VAL = 7
        GO TO 30
C
20      VAL = INCR
30      CHAINS(NTHCHN) = CHAINS(NTHCHN)*8+VAL
        JTHCHN = JTHCHN+1
C*
C*   CALCULATE RUNNING TOTALS FOR LASTPT
C*
      INCVAL = VAL
      DO 40 K1=1,2
        M1 = 3-K1
        CHXY(M1) = 0
        IF(INCVAL.GT.4) CHXY(M1) = -1
        IF(INCVAL.GT.0.AND.INCVAL.LT.4) CHXY(M1) = 1
        INCVAL = MOD(INCVAL+2,8)
40      CONTINUE
        LAST(X) = LAST(X)+CHXY(X)*CHLEN
        LAST(Y) = LAST(Y)+CHXY(Y)*CHLEN
        IF(JTHCHN.NE.6) GO TO 100
        JTHCHN = 1
        NTHCHN = NTHCHN+1
        CHAINS(NTHCHN) = 0
50      CONTINUE
100     CONTINUE
        RETURN
        END

```

Appendix Five

This appendix uses the diagram below to demonstrate the application of the procedure presented in section 6.3.2.

STEP

- (a) all points are standardised in the diagram, hence

<u>Section</u>	<u>Points</u>
A	1 2 3 4
B	2 3 6 5
C	7 8 9 6 5
D	8 9 11 10
E	13 14 6 9 11
F	13 14 15 16
G	15 16 12 4

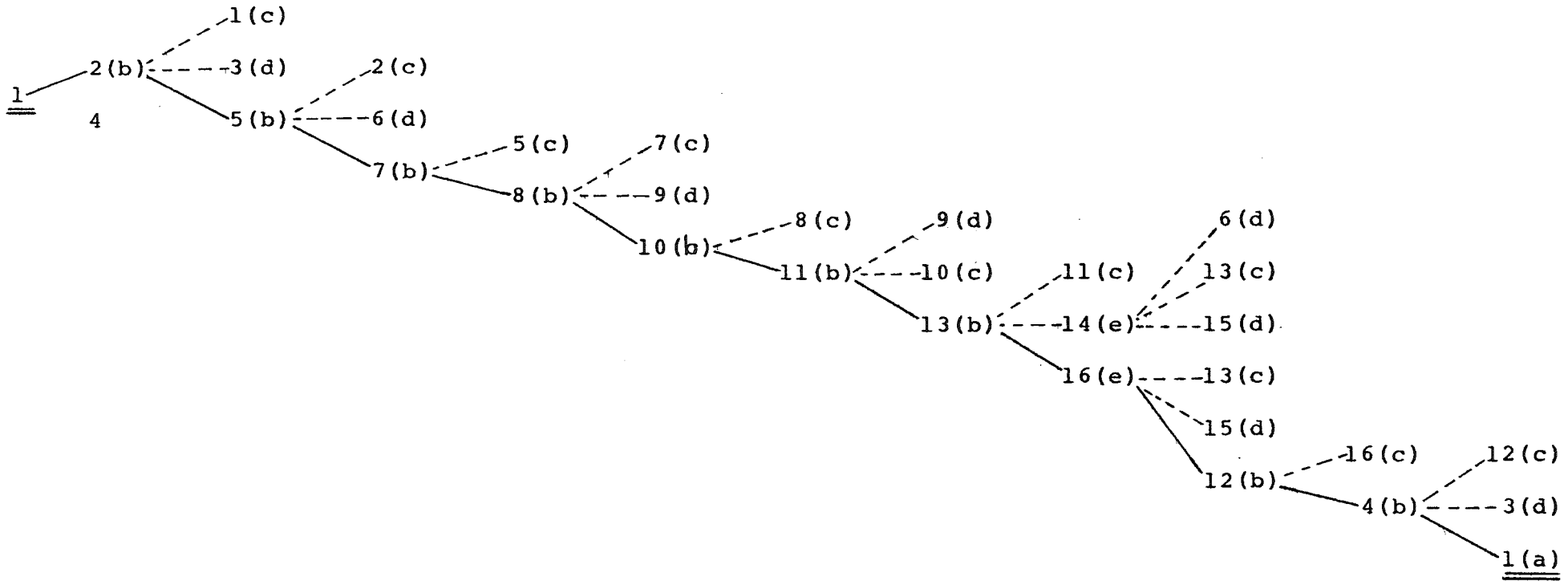
- (b) $S=\{1,5,7,12\}$ These points have the maximum and minimum x and y values from the set of eligible points.
- (c) Points referenced only once, and the points connected to them, are:

4	1	2
5	7	8
8	10	11
13	12	4

Hence $S=\{1,2,4,5,7,8,10,11,12,13\}$

- (d) Starting point is 1.

- (e)-(f) The diagram on the following page gives the possible points on the path at each stage, and for each point, the condition from that algorithm that applies.



Condition

- (a) termination condition. The point is in S, all points in S have been processed, and the point is the starting point.
- (b) point is in S and has not yet been processed, so this point is a required point.
- (c) point is in S but has previously been used, so is not eligible again.
- (d) point not in S, but there are other possible paths to try at this stage.
- (e) point not in S and no other points exist, so try all paths from this point.

Appendix Six

The 1:10000 scale base map on the next page was produced by the MAPPAK system. The map took approximately 40 minutes to digitise and required no editing after digitising. The plotter took 4 minutes to reproduce the XY DICTIONARY data and another 4 minutes was required to plot the CHAIN DICTIONARY for this map. A further 47 minutes were required to add the DDS to the map using the standard character set of the plotter. At the time when this map was produced, the plotting subsystem did not contain the character set described in section 5.3.4.