RIVER BASIN SURVEILLANCE USING REMOTELY SENSED DATA

A Water Resource Information Management System

JAMES DOMINIC FLACH

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

September 1989

The University of Aston in Birmingham

RIVER BASIN SURVEILLANCE USING REMOTELY SENSED DATA

A Water Resources Information Management System

James Dominic Flach

Doctor of Philosophy

1989

This thesis describes the development of an operational river basin water resources information management system. The river or drainage basin is the fundamental unit of the system; in both the modelling and prediction of hydrological processes, and in the monitoring of the effect of catchment management policies. A primary concern of the study is the collection of sufficient and sufficiently accurate information to model hydrological processes. Remote sensing, in combination with conventional point source measurement, can be a valuable source of information, but is often overlooked by hydrologists, due to the cost of acquisition and processing. This thesis describes a number of cost effective methods of acquiring remotely sensed imagery, from airborne video survey to real time ingestion of meteorological satellite data. Inexpensive micro-computer systems and peripherals are used throughout to process and manipulate the data. Spatial information systems provide a means of integrating these data with topographic and thematic cartographic data, and historical records. For the system to have any real potential the data must be stored in a readily accessible format and be easily manipulated within the database. The design of efficient man-machine interfaces and the use of software engineering methodologies are therefore included in this thesis as a major part of the design of the system. The use of low cost technologies, from micro-computers to video cameras, enables the introduction of water resources information management systems into developing countries where the potential benefits are greatest.

Key Phrases: Remote Sensing

Spatial Information Systems

Water Resources Management

Videography

Acknowledgements

The author wishes to acknowledge a number of people for the useful advice and services offered during the course of the research.

Members of staff and postgraduate students of the Remote Sensing Unit and Department of Civil Engineering, in particular Dr.T.Chidley and Dr. W.G.Collins.

Neil Trigwell for his involvement in the airborne video survey work, including the Taw-Torridge survey.

Sue White and the Overseas Development Unit of Hydraulics Research Ltd. for advice in the development of the system and provision of data for the Magat Catchment sample application.

Contents

3 Technological Considerations

8 System Applications

Figures

Tables

8 System Applications

1 RIVER BASIN SURVEILLANCE USING REMOTELY SENSED DATA

1.1 Introduction

Water is essential for life to exist, yet at extremes of quantity: in drought and flood, it can bring death and destruction. It is the role of the engineer in water management to limit the consequences of these extremes and to harness the forces of nature for the good of man.

The current technological advances in digital databases, remote sensing, communication systems and computer modelling, provides the river basin manager with a wide range of possible tools to aid in the analysis of past events, the monitoring of current situations, and the forecasting of future events.

Until recently the civil engineering industry had little or no experience of computers. Computer systems were acquired on the advice of computer scientists and equipment manufacturers. Frequently the systems acquired had the best technical specification - on paper - and were consequently the most expensive, without necessarily being the most suitable. The initial cost of acquiring these systems, and developing them to perform the required tasks, was often high and the benefits were not always immediately realised within a project time scale. Early systems proved unreliable, were costly to maintain, and required staff with an advanced level of expertise in computer systems to maintain, program and operate the system. The technology was still in the hands of the computer scientists.

The resistance to use of computer systems by project managers is slowly being eroded. Development of more efficient processes to produce electronic components and the subsequent increase in the availability of the components has resulted in a rapid decrease in the cost of computer systems, and the advent of affordable desk top micro-computer technology. Over recent years the cost of micro-computers has dropped to the order of £100s, and the available desk top computing power has increased up to the development of the multi-tasking, super-micro-computer, parallel processors and the application specific processor. The super computer of yesteryear is the desk top computer of today.

For many years the IBM Personal Computer has been a standard in micro-computers to be cloned, and improved on, by numerous manufacturers. The popularity of PCs has been accompanied by a growth in the number of interfaces and expansion systems designed for integration with PCs. The PC can now be seen as the starting block for development of sophisticated computer systems. Peripherals and internal expansion can provide bulk data storage, expanded memory, parallel processors (transputers), network communications, satellite data reception, digital signal processing, image processing, graphics, digitising systems, interactive devices, printers and plotters.

The availability of low cost computer systems and their introduction into the home and office, has been reflected by a recent increase in engineering research funding for development of applications and the man machine interface, SERC (1989). The software engineering process has been slow to react to the advances in technology, and consequently a large number of current computer systems are essentially unusable. The design of the user interface is now, justifiably, becoming one of the major concerns of project specifications.

In line with the overall aims of this thesis: to produce a low cost information management system to analyse, monitor and forecast hydrological events, a number of remote sensing systems were considered. The acquisition of data should ideally be under the control of the water management engineer, and therefore the only remote sensing systems considered for multi-temporal monitoring were these that offer real time acquisition of data. This includes global or regional scale imagery, transmitted by meteorological satellites (such as Meteosat and NOAA), and large scale survey data using multiband airborne video. Systems such as Landsat and SPOT are both expensive and inappropriate for high frequency multi-temporal monitoring.

This thesis utilises a number of these technologies in development of a substantial component of an operational water resources information system. All of the processes involved in the production and application of the water information system comply with the basic requisites of civil engineering systems - inexpensive, reliable and user friendly. The system therefore includes elements of micro-computer based graphics, computational and data entry systems, a study of the principles of Software Engineering, and in particular the design of a Man - Machine Interface.

1.2 River Basin Management

In the hydrological cycle one of the most fundamental units is that of the river or drainage basin. Within the river basin there is a balance in the processes of precipitation, interception, soil moisture and groundwater storage with evapotranspiration and runoff, thus maintaining the hydrological unity of the basin. Within the drainage basin the hydrological and geomorphological features interact, forming complex interrelationships. In the management of resources within the drainage basin these relationships are normally modelled by empirical or relatively crude theoretical methods, taking into account the necessary control for unity of the basin. Calibration of models must be achieved by measurement of hydrological parameters; conventionally this is both costly and time consuming. Hydrological models, in common with other physical processes, also suffer from problems of scale, particularly in time step.

The overall problem lies in the collection of sufficient and sufficiently accurate information to enable these processes to be analysed and adequately modelled. Hydrologists, whilst recognising the potential of remote sensing as an input to modelling, continue to use conventional methods of measurement. This is compounded by measurement and modelling of processes without accounting for spatial distribution within the drainage basin. For example, precipitation is commonly measured using sparse networks of rain gauges, generally positioned in easily accessed locations.

The use of remote sensing in hydrology can seen to have three possible applications, Anderson (1979):

1. Replacement of conventional methods of measurement.
2. Combination with conventional methods.
3. Application to problems uniquely solved by remote sensing methods.

The effective replacement of conventional methods is impractical, due the current global usage of existing methods. However, the combination of conventional with remote sensing methods, aided by the use of spatial information systems has many practical applications. For example, remote sensing could provide a basis for spatial interpolation of rain gauge measured precipitation. The analysis of the distribution of soil moisture could also be enhanced by use of remotely sensed measurement, calibrated by conventional ground based methods, and integration into an information system to define the spatial extent of contributing factors such as soil type and land cover.

The limited accuracy of remote sensing methods can be paralleled by the questionable accuracy of conventional methods, which can often be as low as 50%. In developing countries, where there is often few data available, remote sensing can provide a major source of relevant information. In far too many cases hydrologists are overlooking remote sensing, choosing instead to improve the accuracy and utility of conventional measurement by increasingly sophisticated mathematical methods, Schultz (1988).

This shortage of relevant data has led to the predominance of 'lumped catchment', over 'distributed catchment', modelling systems. Lumped catchment systems model the hydrological processes as a single unit over the entire catchment. In contrast, distributed systems model the processes on a grid cell or sub-catchment basis within the basin, providing flexibility of scale. The main problem with distributed models is that they require large amounts of input data. In the past there has been a distinct shortage of spatial data; but with spatial information systems and remote sensing this is no longer the case. There is now an excess of data available for distributed models. Spatial information systems can provide the necessary tools for storage, retrieval and manipulation of spatial data from remote sensing and digital mapping, Ragan (1979), Allewijn (1988), making distributed modelling systems both practical and desirable.

The advent of remote sensing and spatial information systems has significant implications in river basin and watershed management, and provides the capability to monitor the spatial distribution and effect of management policies on hydrological processes. This in turn provides an improved accuracy of predictions, through a more thorough understanding of the spatial variation of hydrological relationships. This should result in more efficient and reliable river basin management projects.

1.3 Data Collection

Adequate data collection from accurate measurement systems, is one of the most important concerns of the river basin manager. The data collected is subsequently used by any number of monitoring, modelling and predictive systems, all of which rely on both accurate and timely information.

Conventionally data collection is achieved using a number of systems from radio and telephone network to site visits. The integrated collection of point source data using satellite telemetry is therefore a useful enhancement of existing methods of data collection, particularly in areas of rough terrain. The late 1980s represents the advent of the Data Collection Platform (DCP) explosion. DCPs provide a method of accessing point source measurement, recorded by a variety of sensors, virtually anywhere in the world. A DCP consists of an array of sensors, recording data, triggered by an internal clock or an event sequence. The data is transmitted to the user via the communications channels on current meteorological satellites, such as the network of geostationary meteorological satellites ringing the globe. These data along with low resolution meteorological satellite image data, can be received and processed on an inexpensive desk top micro-computer.

The integration and analysis of telemetered and image data, transmitted from both geostationary and polar orbiting meteorological satellites, provides a possible source of regional scale, multi-temporal data for monitoring change over the drainage basin. The European Space Agency and NOAA, have made a commitment to maintain these services in the future, enabling operational monitoring systems to be developed.

The use of earth resources satellites, such as the Landsat series and SPOT with ground resolutions as high as 10m, provide a means of producing spatial databases of land use, topography and stream networks as a basis for hydrological modelling, Mauser (1984). Several other hydrological parameters, such as snow cover and soil moisture can also be derived from this technology, for which a number of documented studies exist, Engman (1984), Deutsch et al (1981). The high cost of acquisition, the long time lag between image acquisition and procurement, and the limited repeat cycle, precludes the use of these data for most hydrological applications, particularly for multi-temporal studies. For the majority of applications remote sensing systems do not yet provide sufficient spatial and temporal resolution to adequately monitor hydrological parameters in space and time. Table 1.1, reproduced from an ESA report on remote sensing in hydrology and water management, Herschy et al (1988), details requirements for hydrology with respect to both present and future remote sensing systems.

Aerial survey methods provides sufficiently high resolution data for monitoring and analysis. However the overall cost of acquiring aerial photography, combined with the time lapse between acquisition and analysis, and the subsequent cost of analysis, makes this form of survey extremely expensive for multi-temporal analysis. Although the aerial photograph is best suited for large scale photogrammetric survey, the application of airborne video imagery is worth considering in terms of its in-flight assessment, immediate post flight analysis and its suitability for integration and processing in digital image processing and information management systems.

Descriptions of the physical basis of remote sensing, sensor systems and the possible applications of these data, will not be discussed in this thesis, but can be found in any of a number of excellent texts on the subject, Colwell (1983), Lillesand and Kiefer (1979), Cracknell (1981).

Parameter	Resolution	Frequency	Accuracy
A. Precipitation	1 km	60 min	20%
B. Snow - Land			
Depth	200 m	24 hr	50 mm
Snowline	1 km	24 hr	2%
Snowcover	1 km	24 hr	5%
Water equivalent	1 km	24 hr	10 mm
Free water content	1 km	24 hr	2%
Snow surface temperature	1 km	24 hr	0.5 C
Snow albedo	25 km	7 day	5%
C. Lake & River Ice			
Ice line	30 m	12 hr	2%
Continuous ice cover	25 m	24 hr	10%
Ice concentration	30 m	24 hr	2%
Ice movement	30 m	24 hr	20 m
Thickness	25 m	24 hr	10%
Surface temperature	25 m	24 hr	1.0 C
D. Glaciers			
Inventory	25 m	1 yr	2 %
Snow cover	25 m	7 day	2 %
Variation - length	30 m	6 month	30 m
Mass balance	1 km	30 day	10 mm
Surge monitoring	25 m	7 day	10 %
E. Surface Water			
Areal extent	30 m	24 hr	3 %
Saturated soil area	30 m	24 hr	3 %
Flood extent	100 m	12 hr	2 %
Flood plain	20 m	3 year	2 %
Lake / river stage	20 m	1 hr	50 mm
Waves, seiches	20 m	1 hr	200 mm
F. Groundwater			
Aquifer maps	100 m	3 year	10 m
Discharge local rivers	30 m	7 day	10 m
Discharge local lakes	50 m	7 day	10 m
Location springs	15 m	2 year	5 m
Groundwater level	100 m	7 day	5 mm
Soil type	100 m	5 year	
Moisture content	100 m	7 day	5%
Temperature profile	500 m	24 hr	0.5 C
Infiltration	300 m	3 day	10%
Percolation	300 m	3 day	10%
Frost depth	300 m	3 day	10%
Permafrost area	300 m	3 year	5%

Table 1.1 (continued over page)

Parameter	Resolution	Frequency	Accuracy
G. Evaporation			
Evaporation	1 km	24 hr	20%
Evapotranspiration	1 km	24 hr	20%
H. Water Quality			
Turbidity	100 m	6 hr	20%
Suspended solids	100 m	6 hr	20%
Colour	100 m	6 hr	
Algal blooms	100 m	12 hr	20%
Surface films	100 m	6 hr	20%
Surface water temperature	500 m	7 day	0.5 C
Temperature profile	5 m	15 day	0.5 C
I. Basin Characteristics			
Drainage area	20 m	5 year	0.5%
Channel dimensions	10 m	3 year	3%
Overland flow length	10 m	3 year	3%
Surface slope	10 m	3 year	3%
Land cover type	200 m	6 month	3%
Albedo	50 m	6 hr	5%

Table 1.1 Observational Hydrological Requirements for Satellite Remote Sensing Systems, Herschy et al (1988).

1.4 Spatial Information Systems

The spatial information system handles spatially located and related data. It is analogous to a Geographic Information System (GIS) geared towards engineering rather than geographical applications. GIS technology is a growth field in research, development and applications in resource mapping, NERC (1988). The nature of GIS includes support for both temporal and spatial dimensions, making it an ideal tool for multi-temporal monitoring and analysis. In terms of hydrological forecasting and modelling, the water information system provides a system for handling real time acquisition of telemetered point source data, meteorological satellite image data, earth resources satellite image data, aerial survey data, digital map data, cartographic map data and statistics.

The basis for processing information for modelling is a spatial database, generated from large scale remote sensing survey, and cartographic data. Digital map data are rarely available outside the developed world, and therefore the production of base themes within the spatial information system: such as land use, topography, soils, geology, elevation, slope and aspect, is often the most time consuming stage of the project. Methods chosen must therefore be efficient, reliable and user friendly. Methods for the production of spatial databases detailed in this thesis uses inexpensive desk top micro-computer technology, video digitisation and document scanners.

To a large extent the hardware technology of the information system display controls the nature of the processing of the spatial data. Being an image processing based system, spatial data is displayed as a raster. Digital map data are stored and processed in vector format until areal or spatial relationships are studied, whereupon the vector is converted to its raster/pixel representation.

One of the most useful database themes and input to modelling systems is the incorporation of the third dimension - the Digital Elevation Model (DEM). The production of DEM from cartographic data, involves a number of complex and computationally expensive routines, from contour extraction to interpolation procedures. In addition there is a wide range of products derived from DEM, such as slope, aspect and watershed delineation, all of which have obvious applications in modelling systems.

In developing countries spatial information systems can form the basis for the management of data collection, integration of data, interpretation and map production. Systems such as the Integrated Land and Watershed Management System - ILWIS, Meijerink (1988), developed by ITC, Netherlands, have already been used by mapping agencies in the Far East. In comparison UK foreign aid has been slow to respond to the requests for supply of remote sensing and information systems, leaving unsupported UK companies to compete with Canadian, German and Dutch government aid projects. The introduction of desktop and portable micro-computer based systems, and the development of efficient user interfaces, make this technology available to both UK consultants and foreign governments with limited budgets.

2 SPATIAL INFORMATION SYSTEMS

2.1 Definitions

Spatial Information Systems are part of the total Information Technology that includes on-line data retrieval and processing. The term *Spatial Information System* is synonymous with *Geographic Information System* (GIS) and *Spatial Analysis System*. Throughout this thesis Spatial Information System and Water Information System have been used to indicate an information system oriented towards applications in engineering, particularly water resources management. This system integrated with real time acquisition of remotely sensed data forms the basis of the river basin surveillance system. All spatial information systems are integrated by their very nature and consequently *integrated* is often used in description of information systems, for example, Integrated Spatial Information System (ISIS), Integrated Geographic Information System (IGIS)

Geographical Information Systems have been variously described as:

"A spatial data handling system", Marble (1983).

"...information system that handles geographically or spatially referenced data", Parker (1988).

"...a powerful set of tools for collecting, storing, retrieving at will, transforming, and displaying spatial data from the real world...", Burroughs (1986).

"A system for capturing, storing, checking, integrating, manipulating, analysing and displaying data which are spatially referenced to the Earth. This is normally considered to involve a spatially referenced computer database and appropriate applications software", HMSO (1987).

The GIS may comprise a number of integrated systems dedicated to an individual process and may therefore be described as a technology rather than a system.

> "...an information technology which stores, analyses, and displays both spatial and non-spatial data", Parker (1988).

The common feature that differentiates spatial information systems is the ability to process spatial data in addition to numerical and textual data. Spatial data processes involve space, a spatial data element having physical dimensions. Each feature is composed of a collection of basic spatial components - points, lines and polygons. It is described by its location in geographical space, its attributes or descriptors, and its spatial relationship with other features - topology.

In this context remotely sensed data is just one of many sources of spatial data. Additional data sources include digital databases, existing cartographic data, tabular data and statistical data. A spatial information system must therefore be capable of handling these as well as a variety of other data.

There is a certain amount of overlap between GIS, CAD, digital mapping and database management systems (DBMS). On closer examination, a number of commercially available GIS bear closer resemblance to their individual components. Perhaps the most fundamental requirement of these systems, and consequently the factor discriminating CAD from DBMS from GIS, is the ability to create entirely new data sets from existing data sets; such as land use mapping from remotely sensed data.

2.2 Components

A Geographic Information System can be described as a collection of technologies, and may consist of a number of separate systems linked via common formats. For example a system performing all the functionality of a GIS could use a CAD system for digital mapping, and a relational database management system for storage, intelligent retrieval and creation of new data. Many current systems adopt this format, linking digital map data to relational database management systems such as ORACLE or INGRES. There are a number of problems associated with use of these, and similar proprietary databases for spatial information systems. Although most DBMSs can provide the functionality, very few can handle and manipulate spatial data and topological relationships efficiently. For this reason, systems dedicated to the retrieval and processing of spatial data prove more efficient and flexible.

2.2.1 Design Philosophy

There are a number of design philosophies for integrated information systems: all must provide the basic capabilities of a GIS to facilitate the capture, storage, retrieval, analysis and display of spatial data. These approaches to design can be described as Process Oriented, Application Oriented, Toolbox and Database approaches, Cowen (1988).

1. The process oriented approach relies on integration of several sub-systems to provide the total solution, such as CAD and DBMS systems.

2. The application oriented approach is a system specifically designed to perform only those functions required necessary for a particular application. The system may not require all the functionality of a GIS, but performs the application more efficiently.

3. The toolbox approach provides the functionality of the full GIS through a set of computer based procedures and algorithms. Each function is accessed via a menu system or command line system. Applications oriented systems may be developed from a toolbox using batch processes.

4. The database approach emphasises the requirement for ease of integration of spatial data manipulation procedures in a relational database management system. The DBMS provides the tools for intelligent query and retrieval of spatial data. Standard DBMS are not designed for manipulation of spatial data, consequently substantial research is being undertaken in development of the optimal spatial database management system.

2.2.2 Spatial Data Base Management Systems

A Spatial DBMS is specifically designed for the rapid, intelligent retrieval of sufficient spatial data to generate a thematic map. The major advantage of using a DBMS is that the processing of the data is separated from the storage and retrieval of information. The DBMS standardises on data access independent of physical storage structure, provides the interfaces to the operating system and enables multi-user access via a high level language or query.

Faster retrieval of data is aided by optimal data storage structures and file processing operations, Frank (1988). Fast data access generally relies on minimisation of disk drive head seek time. This can be achieved by use of clustering and buffering techniques. Clustering groups data by geographical location into pages, which are read into buffers from which the relevant data is extracted. To aid this, geographical location query is generally reduced to rectangular areas.

With large national databases, the DBMS handles transaction processes and multi-user access. Individual transactions can be logged and checked for accuracy, concurrency and integrity before update of the database, and the database protected against illegal entry.

The retrieval of thematic spatial information can also be enhanced by the addition of Knowledge Bases, Usery (1988), implemented within the DBMS.

2.3 Spatial Data Representation

A Spatial Information System handles three main types of information; locational, non-locational and temporal.

1. Locational data can be described in terms of a measured set of co-ordinates, such as points, lines and polygons; or can be described topologically, such as a grid or network.

2. Non-locational data can be a variable, class, value or name, that has no geographical location, but describes a spatial feature.

3. Temporal projects spatial data into the temporal dimension, for analysis of change over a period of time. Temporal data can be both locational or non-locational.

Permanent	Locational	Measured x-y	Point Line Polygon
Temporal		Topological	Grid Network
Permanent	Non- Locational	Variables Classes	
Temporal		Values Names	

Table 2.1. Spatial Data Types, Dangermond (1984).

Geographically located data can be further subdivided into seven main data types, each of which can be represented in terms of points, lines and polygons. This is pictorially represented in Figure 2.1.

At this stage two main methods for the analysis of spatial data can be described: that of network topology or grid cell topology. The choice of method depends to a large extent on the application, the range of possible inputs, the available computer graphics hardware and the data storage requirements.

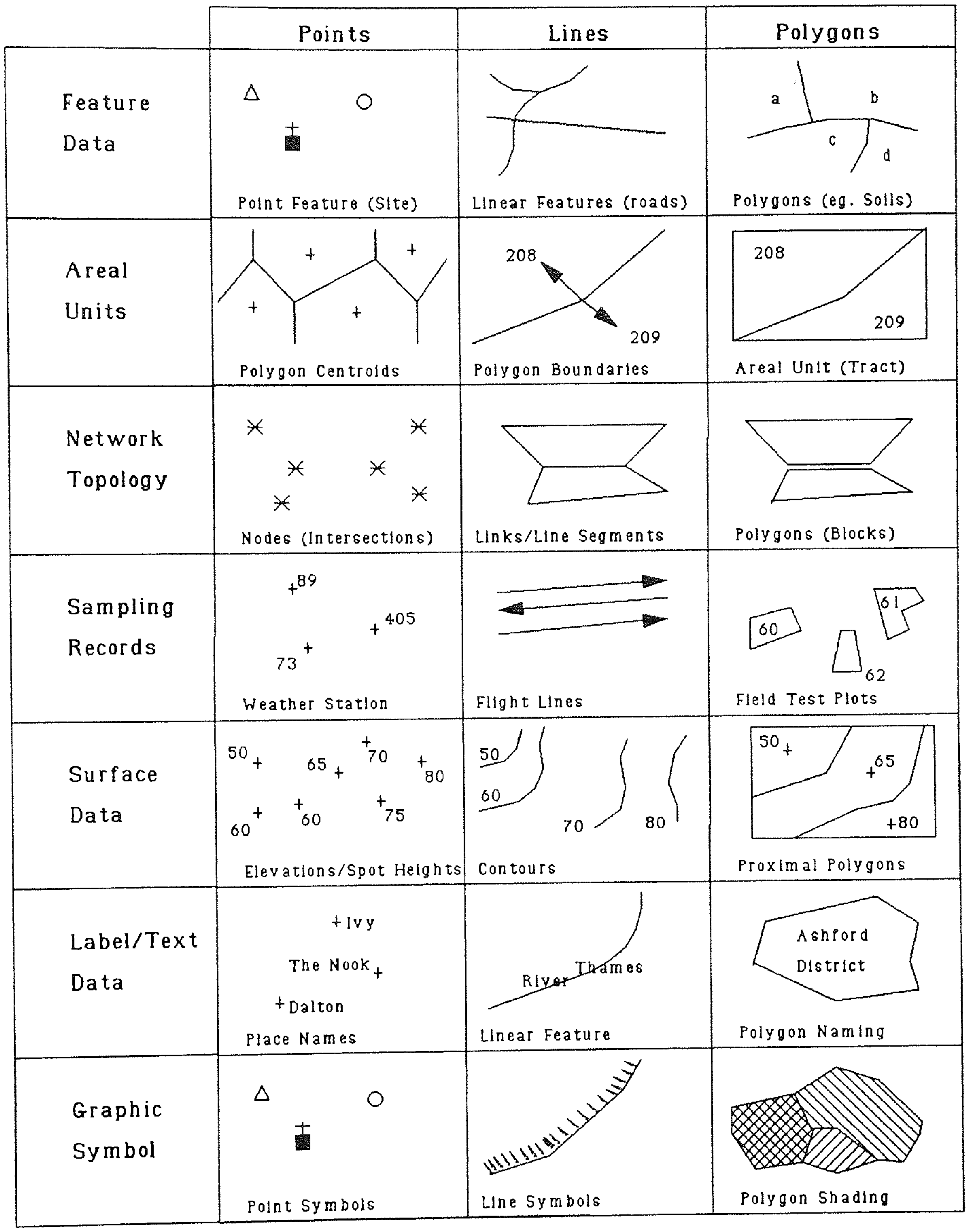


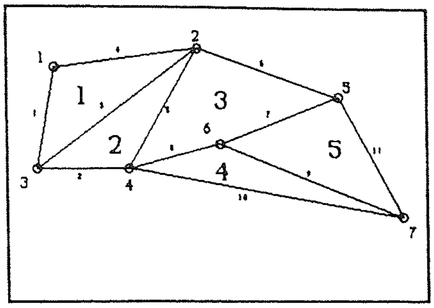
Figure 2.1. Geographic Data Types and Representation, Dangermond (1984).

2.3.1 Network Topology

Network based systems rely on the manipulation of vector information in a graphics based system. The thematic map is represented by vector draw and polygon fill functions, using the network node as input. A polygon is defined by a set of lines connecting boundary vertices and a boundary or seed fill process. Network based systems allow more flexible scaling and are more efficient for storage of sparse networks.

The data stored in a network based system can be structured in a number of ways. Points and lines are represented by a single co-ordinate or list of co-ordinates. However the structure of polygon storage is more complicated.

The simplest structure relies on each polygon being complete. With adjacent polygons this would require each boundary segment to be repeated. More complicated structures, such as that described by Dangermond (1984) and Haralick (1979), define a polygon as a number of line segments connecting network nodes. Each line segment is related to the polygons to the 'left' and 'right'. This form of data structure is used by a variety of commercially available GIS such as ARC/INFO and SPANS, and is represented in Figure 2.2.



Link	Right Polygon	Left Polygon	Node 1	Node 2
1	1	0	3	1
2	2	0	4	3
3	2	1	3	2
4	1	0	1	2
5	3	2	4	2
6	3	0	2	5
7	3	3	5	6
8	4	3	6	4
9	5	4	7	6
10	4	0	7	4
11	5	0	5	7

Node	X Co-ord.	Y Co-ord.
1	23	8
2	17	17
3	29	15
4	26	21
5	8	26
6	22	30
7	24	38

Figure 2.2. Network Based Polygon Representation.

2.3.2 Grid Cell Topology

Grid cell, or raster, based systems transfer point, line and polygon data to a grid, where the value stored in each element of the grid (pixel) relates to a particular polygon. Polygons are therefore defined by a number of connected pixels of a single value. Grid cell systems are closely related to remotely sensed data, and can therefore be integrated and manipulated using framestores and image processing functions. Raster based systems facilitate analysis of spatial and areal variation and are more efficient for storage of dense networks.

The data structure of a grid cell based system consists of sequential storage of data representing class data, related to methods of raster scan of the grid cell. Points, lines and polygons are 'scan converted', Foley and Van Dam (1984). A point is then represented by a single grid cell located at the integer representation of the co-ordinate; a line by a series of cells, described by functions such as Bressenham's line drawing algorithm; and a polygon by a block of cells, consisting of a set of horizontal scan lines. This is represented by Figure 2.3.

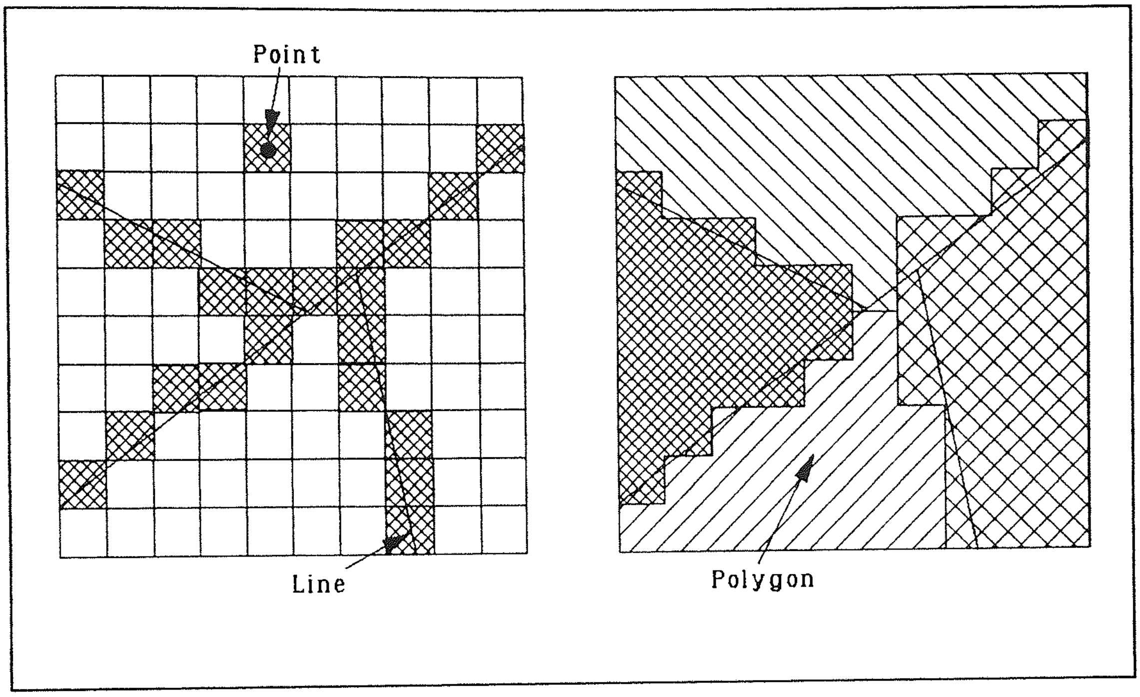


Figure 2.3. Grid Cell Based Feature Representation.

Storage of raster data can also be reduced by use of compression techniques such as simple run-length encoding or more complex methods such as quadtrees. The storage savings are substantial for these data, but retrieval time is affected.

2.3.3 Combined Representation

Both types of system must provide utilities for conversion; such as pixel representation of lines, and extraction of vertex data from a grid cell polygon. This forms part of the basic functionality of any spatial information system.

Experience has shown that systems combining the functionality of both grid cell and network based systems have the greatest utility, and provide for more efficient storage and manipulation of data.

The example system, developed as part of the water information system, has support for vector or network based capture, edit, storage and manipulation of data on a raster based graphics system. Vector manipulation is the optimum method for handling digital map data such as linear features and boundaries. Areal features are implicitly defined by a collection of line segments, but not necessarily formed on retrieval. Analysis of spatial relationships and overlaying of data sets is performed on a grid cell basis after retrieval. Thus class maps are stored as vector boundary data and converted to grid cells prior to analysis. This process is more fully described in Chapter 4.

2.4 Functionality

The functionality of a spatial information system is best described by a definition of the procedures involved. A number of systems have been documented, including ARC/INFO, Dangermond and Friedman (1984), and the SPANS, marketing literature. The functionality can be classified into 3 main components:

1. Data Capture
2. Data Editing
3. Data Analysis

2.4.1 Data Capture

Data capture must allow for input of digital and non-digital, areal, point source and temporal data. Integration of digital data is a function of data transfer between formats. Support must also be provided for input of cartographic data. This particular function is perhaps the most time consuming and costly stage of a project, and consequently any method of increasing efficiency must be investigated and evaluated.

The problem is one of digitising typically large volumes of spatial data, across several map sheets, maintaining the spatial relationships and attributes of the features, Calkins (1984).

The aim of any digitisation process is to extract features into a database with sufficient accuracy that its use is efficient. The magnitude and frequency of errors should therefore be limited by the controlling process. Accuracy of digitisation systems depends on a number of factors, Cameron (1984). These include defects in the geometry and mechanics of the system and more prominently human errors. The man machine interface to digitisation systems is therefore important in making errors easily detectable and edited. All digitisation systems must provide support for subsequent editing of data prior to final inclusion in the database.

2.4.2 Data Editing

Data editing procedures make use of the graphic display of the system to highlight errors produced by the capture process. Errors in spatial location of features are handled using on-screen edit of a spatial display.

Typical functions include:

1. Display of digitised/coded data for visual editing.
2. Topological relationship checks for redundancy, repetition and conflict to ensure correctness of data.
3. Removal of erroneous features and line segments.
4. Polygonisation of line segments, join and split functions for grouping of data.
5. Editing of feature vertex co-ordinates.
6. Edge match analysis to ensure smooth transition across map sheet boundaries.

2.4.3 Data Analysis

Data analysis forms the largest specification of the functionality of a spatial information system, and provides the utilities for creating new and useful thematic products from captured data. The functionality, essentially describing ARC/INFO, a network based GIS, can be further subdivided into the following categories.

1. Data Retrieval (see 2.4.3.1)
2. Map Generalisation (see 2.4.3.2)
3. Map Abstraction (see 2.4.3.3)
4. Map Sheet Manipulation (see 2.4.3.4)
5. Buffer Generation (see 2.4.3.5)
6. Polygon Overlay (see 2.4.3.6)
7. Measurement (see 2.4.3.7)
8. Grid Cell Functions (see 2.4.3.8)
9. Digital Terrain Modelling (see 2.4.3.9)
10. Output Techniques (see 2.4.3.10)

A number of database techniques are also described for manipulation of locational data and provision of the operating system interface. These features can be performed by DBMS geared towards spatial data. The system also makes use of the relational aspects of the DBMS in attribute aggregation prior to retrieval and query.

1. Base File Creation and Update
2. File Management
3. Search, Retrieval and Display
4. Query of Selected Attributes
5. Query of Selected Attributes by Geographic Location
6. Window Manipulations
7. Edge Matching of Map Sheet Data
8. Conversion of Data Formats

In addition a number of functions related to CAD systems are utilised for graphic manipulation, at stages subsequent to data capture.

1. Text Labelling of Data
2. Creation of Symbol Libraries
3. Storage of Symbol Libraries for Recall and Display
4. a) Windowing, Enlarging and Symbol Manipulation
   b) Editing of data elements
   c) Graphical Compositing of Spatial Data Sets
   d) Overlay of Geographical Reference Grid
5. Interactive Entry of Data

2.4.3.1 Data Retrieval

The intelligent retrieval of data from the database involves a number of functions for the combination of multiple thematic maps using boolean operations, locational and attribute query and windowing of data.

Functions include:

1. Browsing for 'quick look' of graphical and tabular data
2. Windowing by geographical location
3. Query Windowing
   a) Adjacency Analysis
   b) Point and Polygon retrieval within window
   c) Polygon Overlay within boundary
4. Multiple map sheet spatial query
5. Boolean operations for retrieval, combination and aggregation

2.4.3.2 Map Generalisation

Map generalisation is a process of reducing the data necessary to represent the spatial location of features. For linear features this involves the removal of vertices that do not diverge from a straight line joining vertices. This has the effect of reducing the number of line segments required to adequately represent the spatial location of a feature. Polygon generalisation involves a linear generalisation of line segments and the removal of line segments that separate two polygons of the same class.

Functions include:

1. Line co-ordinate thinning
2. Polygon line segment co-ordinate thinning
3. Polygon boundary segment redundancy
4. Edge matching functions

2.4.3.3 Map Abstraction

Map abstraction involves the derivation of new data based on the spatial relationships of the data. This involves functions to calculate polygon centroids, produce contour plots from random data, analyse proximity (nearness), reclassify polygon and convert network data to grid cell format.

1. Polygon centroid calculation
2. Contouring from random data
3. Proximal mapping
4. Polygon reclassification
5. Conversion to grid data

2.4.3.4 Map Sheet Manipulation

Map sheet manipulation involves the correction and transformation of data extracted from map sheets to account for paper stretch, scale and projection change, rotation, shear and translation. This enables maps of various scales and projections to be integrated and converted between common co-ordinates projections.

1. Scale Change
2. Removal of geometric distortion
3. Change of map projection (Rectangular, Transverse Mercator)
4. Rotation/Translation/Shear of co-ordinate data.

2.4.3.5 Buffer Generation

Buffer generation is similar to proximal mapping in that a buffer or corridor is generated indicating a range of distances from a particular feature. Proximal mapping generates a surface that represents distance to the nearest feature selected and indicates which feature the minimum distance corresponds to. These are known as proximity surfaces and voronoi diagrams. Proximity surfaces yield proximal corridors (buffers) by classification of distance ranges. A special case of voronoi diagrams, applied to particular type of point feature, rain gauges, produces Thiessen Polygons for determination of average rainfall.

Buffer generation determines the loci of points less than a specified distance from a spatial feature, and consequently can be used to determine zones of influence.

2.4.3.6 Polygon Overlay

Polygon overlay is a method of creating new thematic products from existing products. Typical combination operators are boolean and arithmetic. An example application finds the location of the overlap of all areas of one class from one theme with a group of classes from a second theme, eg. all areas, with a land use described as forest, at a height greater than 1000m.

This is typically achieved using a matrix function, whereby the classes of each theme relate the offset into the matrix. Table 2.2 shows a typical matrix overlay function.

GEOLOGY	LAND USE					
	Urban	Grass Land	Cereal Crops	Root Crops	Decid. Forest	Conif. Forest
Sandstone		X				
Siltstone						
Limestone				X		
Basalt						
Granite						

Overlay Function:- Geology = Sandstone AND Land Use = Grassland
OR Geology = Limestone AND Land Use = Root Crops

Table 2.2. Typical Matrix Polygon Overlay.

Conversion to grid cell format prior to polygon overlay functions provides for a more efficient combination of themes, network based system rely on the formation of new polygons defined by the polygon boundaries and intersection points. Both result in the creation of new thematic information, which can be manipulated similarly to the original input data.

2.4.3.7 Measurement

The majority of applications use the created data as input to further analysis. They may require the measurement of features in terms of their numerical, linear and areal extent. Inclusion of a third dimension, such as topographic elevation, enables volumetric measurement of change.

1. Measurement of number of features of a given type and attribute
2. Line length measurement
3. Areal measurement, polygons, polygon overlap
4. Volume measurement

2.4.3.8 Grid Cell Functions

Grid cell analysis provides functionality to convert between topological data types. To some extent this relies on use of a network based structure for the majority of functions. In any system functionality should exist for transference of vector/network data structures. When this function is performed depends on the graphics hardware used. Consequently the majority of functions overlap with a number already discussed, the difference being the use of grid cell data and integer arithmetic.

1. Grid cell overlay, as 2.4.3.6
2. Areal measurement, as 2.4.3.7
3. Search radius aggregation of data to specified resolution
4. Distance calculation, integer based.
5. Optimal corridor selection, as 2.4.3.5

2.4.3.9 Digital Terrain Modelling

The majority of information systems provide support for integration of Digital Elevation Models (DEM). A DEM is simply an array of x-y-z co-ordinates represented by either a network topology, such as Triangulated Irregular Network (TIN), or a grid cell topology. Transference between formats invariably involves interpolation and contour generation.

Typical products created from DEM within the information systems are slope, aspect, shaded relief, visibility, automatic watershed partitioning and viewing transformations. The production of DEM within the water information system, and the range of possible applications is described in Chapter 6.

2.4.3.10 Output Techniques

The aim of information systems is to provide an integrated method of producing new, useful, thematic information. This relies on adequate display and hardcopy facilities, storage and production of tabular statistical data. Selection of devices for production of hardcopy depends on the nature of the data produced, and the graphics hardware used.

The basic choice for hardcopy is between an x-y plotter or a colour line printer. For output of network topology, linear and textual plotters are more efficient. However thematic products produced from class maps, containing substantial areal representation, are best reproduced on a printer.

The same applies to the graphics display system used. Vector based, line drawing hardware should be used for network data, representation of areal features is performed more efficiently on raster graphics systems using a grid cell topology. Graphics display devices and further technological considerations are discussed in Chapter 3.

2.5 Advantages and Disadvantages

The overall advantage of using spatial information system compared to conventional (paper) methods of storage and manipulation, is that all data is collected, integrated, stored, retrieved and manipulated on a common base. Further advantages of using spatial information systems, over conventional systems, can be summarised as follows:

1. The data is stored in a physically compact form, on computer compatible tape (CCT) or other magnetic storage device. The necessity to allocate areas for storage of large volumes of cartographic data is removed.

2. The data can be retrieved, updated and stored at a lower cost per unit of data.

3. The process of retrieval is performed with much greater speed.

4. The measurement, transformation, overlaying, compositing and design of data products is performed using the computer.

5. Statistical, tabular, graphic and non-graphic/attribute data can be merged and manipulated taking into account spatial relationships.

6. Input to conceptual models can be extracted, and altered within the database making rapid and repeated testing more efficient.

7. Multi-temporal change detection analysis can be more efficiently performed.

8. CAD and graphics systems can be used to format output.

9. The production of derived products is more efficient due to the flexibility of the database. For example, production of slope, aspect, shaded relief, watershed from elevation; production of overlay analysis from multiple theme map sheets.

10. The overall system of data collection, spatial analysis and management is performed on an integrated basis.

Its disadvantages compared with conventional systems are mainly due to the initial cost and expertise required in setting up the database.

1. The cost of incorporating existing data into the system is high.

2. A large amount of financial and technical overhead is required to maintain the system.

3. The cost of initial acquisition of the system is high.

4. In certain applications the cost benefits are marginal.

2.6 Typical Applications

The applications of spatial information systems are numerous, but can be classified into certain subject areas. These include:

1. Engineering mapping - production of large scale digital mapping products for engineering and construction applications, such as highway and pipeline route selection, drainage areas and service utilities mapping.

2. Automated photogrammetry - linking stereo plotters and analysers to digital mapping systems for storage and manipulation of extracted data in a database.

3. Cadastral mapping - production of ownership boundaries for land registration in government, taxation, conveyancing etc.

4. Event mapping - for mapping the distribution of statistical measures, such as population, crime rate and frequency of accidents.

5. Land use planning - on both local and regional scales, including environmental impact analysis and management of natural resources.

6. Transportation network analysis - for routing of public service and private vehicle transport.

7. Modelling - generation and extraction of input data for conceptual, geographical, hydrological and hydraulic models.

A cross section through this range of applications is presented throughout this thesis. Chapter 8 details applications of spatial information systems integrated with remote sensing. These applications detail the development of hydrological models, hydraulic models and monitoring systems for regional resource and engineering management.

3 TECHNOLOGICAL CONSIDERATIONS

3.1 Introduction

In the design and development of an integrated information system, the systems designer must first evaluate state-of-the-art technology in terms of both hardware and software. This involves the choice of host computer, dedicated hardware, peripherals, communications, operating systems and application development tools.

The design and development of customised hardware components for the system is both costly and time consuming. The use of off-the-shelf products in the system, with accompanying software support, significantly reduces this effort. It is this approach that has been adopted in development of components of the integrated information system described in this thesis.

It has been suggested that the expense of developing software for systems is currently about 90% of the total system cost. Software engineering is an attempt to reduce cost and increase efficiency of programmers by introduction of programming methodologies.

Research and development in computer science has until recently been geared to improving the efficiency of software and hardware products. The advent of more powerful computers and the increased presence of the computer in industry has caused an increased awareness of problems of human computer interaction. Still in its early stages the design of the man machine interface is largely an art. Several basic principles provide a broad description of the design process.

Although to some extent a large amount of work presented in this chapter will be obsolete in a number of years, particularly with respect to current hardware technology, it is worth reviewing some of the technology in a little more detail.

3.2 Software Engineering

Software engineering was a term first penned in the early 1970s to describe the various design principles covering the engineering of a computer system. It was described by Bauer (1971) as:

> 'The establishment and use of sound engineering principles (methods) in order to obtain economically, software that is reliable and works on real machines'

The process of software design can be split into a number of processes. These processes include the determination of specifications; problem formulation; implementation; testing, operation and maintenance. The process is essentially iterative as the software is gradually refined towards the end product. This process can be described as the Software Life Cycle.

3.2.1 The Software Life Cycle

The description of the software life cycle takes into account the complete process of software design including analysis and design, coding, testing and integration.

Figure 3.1 shows the Software Life Cycle from time of conception up to integration and maintenance.

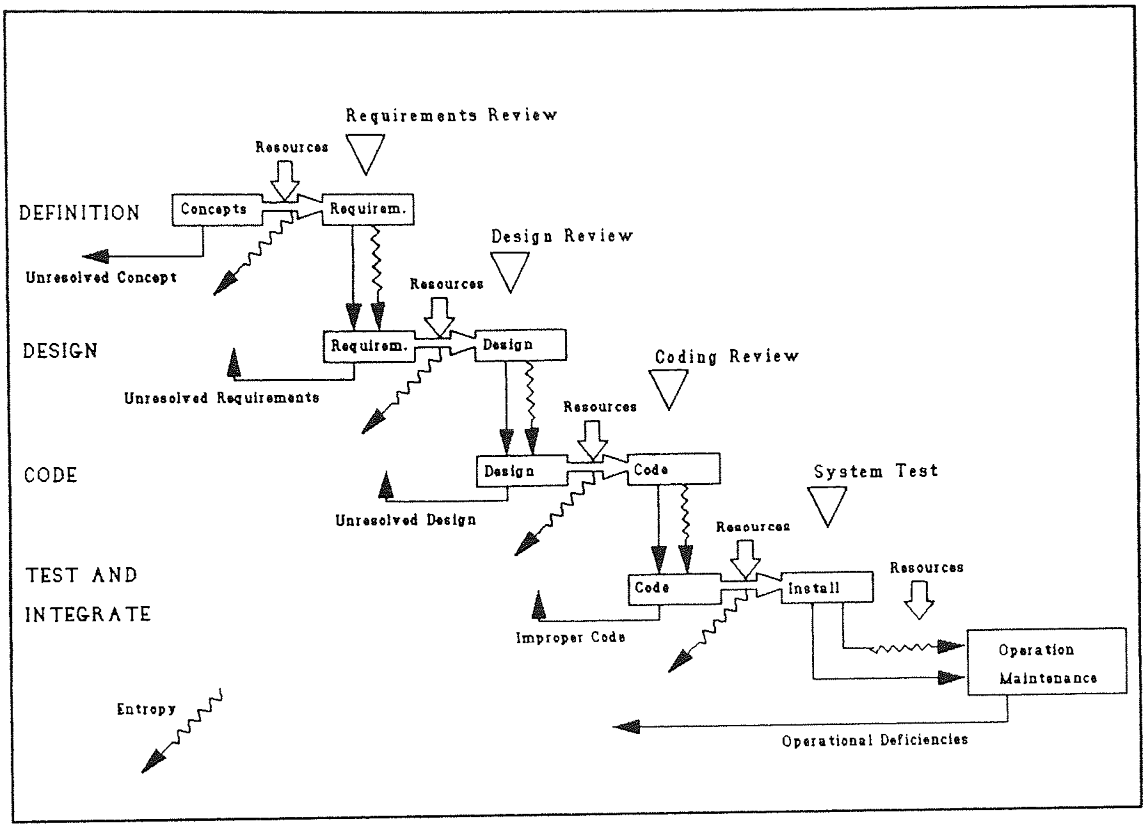


Figure 3.1. The Software Life Cycle, Jensen (1979).

The initial process is the definition of the concepts which control the system. This leads to a specification of the user requirements for the system. At each stage resources are applied and entropy is produced. Entropy, in this context is resource applied with no result.

The requirements lead to the design, followed by the implementation in code. The completed code is integrated in to the final system.

At each stage, including operation and maintenance, the design is reviewed and can fail, requiring the previous step or steps to be repeated.

3.2.2 Software Design

The design process is controlled by 5 major themes describing the problems involved.

1. Software design is a human problem solving process. Human beings solve the problems, machines implement the solutions. The first stage in any problem solving process is to analyse the problem, this can be done only with a precise specification of requirements.

2. The design results in a product designed for use by someone other than the creator of the design. It could be suggested that the majority of systems are based on the creators translation of what the user requires, rather than what the user actually wants. It is obvious that in depth formulation of the problem is required.

3. Software design is closely related to system design, for the majority of systems it is easier to translate requirements into system design using off the shelf products, rather than design all components of the system.

4. All design should be structured. This helps in the control of the production process. With large systems it is easy to lose track of the results. The structured programming process splits the system into a number of individual modules. This helps control and accelerates the process when the modules are spread over a large number of programmers, provided the modules have distinct boundaries, preferably single entry single exit routines.

5. Design verification is a continual process starting with the identification of the problem and a statement of requirements and ending with maintenance and testing.

To summarise the features of a well designed system can be listed, Jensen (1979).

1. Reliability	The system performs to the requirements as to input, output and the process it performs.
2. Testability	The system is easy to test for the user requirements.
3. Maintainability	The system can be easily understood by maintenance programmers.
4. Efficiency	The system performs the process without unnecessary wastage of resources.
5. Understandability	The functionality of the system is easily understood.
6. Adaptability	The system is legible, device independent, structured and self contained.

A further 6 desirable features can be added to the above.

7. Necessity	Only those performance requirements necessary to meet the user requirements are included.
8. Completeness	All modules in the structure are identified, all interfaces specified and all environments noted.
9. Consistency	The design philosophy is compatible throughout.
10. Traceability	All end functions must be related to user requirements.

11. Visibility	All elements must be evaluated to design decisions.
12. Feasibility	The design must be evaluated to be implementable within the constraints of cost and schedule.

3.2.3 Structured Programming

The conceptual view of the system is automatically subdivided by the very nature of human logic into modules that can be separately tackled and understood. This is the basis of the structured programming approach to software design. In this the design is expressed as an hierarchical structure. The structure then decomposes into nodes until a logical termination point is reached. At this point further decomposition of the structure will only add superfluous details and will distract from the understanding of the overall design.

The structure can be designed in 3 ways: from the top down, if the design is a discovery process of the components required; from the bottom up, if logical components of the system exist; or from both ends, if known parts exist and are used to weigh the logic process. As an example of the latter, simple graphics primitives hide many awkward software features caused by the hardware used and should therefore be designed first, followed by a top down design of the functionality of the graphics system.

There are many definitions of structured programming, among the most accurate are:

> 'Structured Programming is the formulation of programs as hierarchical, nested structures of statements and objects of computation', Wirth (1974).

> '...the process of decomposing a problem into successive layers of more manageable or comprehensible pieces that result in a hierarchical structure describing the problem and/or solution in levels of increasing detail', Jensen (1979).

The search for new structured programming methodologies can be attributed to the 'software crisis': the lag between the development of software and computer hardware.

A prime example of the software crisis is the transputer. On its own an impressive processor, but with links, large arrays of transputers can be configured to run concurrently. As of 1989 transputer based systems have not really transferred to the user market, this is due to the cumbersome operating system and low level of support. The development of software for the transputer has two disadvantages. Firstly, until recently the only way to program the processes in parallel was to link processes with Occam code, the transputer language. Secondly, and more importantly the jump from traditional serial coding to parallel coding is large, and can therefore be prohibitively time consuming and hence costly.

The increase in power and available memory at low cost now means that the cost of supplying software for a system can be as high as 90% of the total cost. The time taken on software development is therefore at a premium, and there is an obvious need for new methodologies to increase efficiency and overall productivity.

3.3 The Man Machine Interface

Until recently, the analysis of the human factors, or ergonomics, of computer system design has been largely ignored. The majority of effort expended in the design and operation of computer systems has been in the production of efficient programming styles - software engineering. This has been due to the limitations of computers in terms of available memory and processing power. Over recent years the introduction of low cost, powerful, desktop computers into the office and home, has highlighted the need for considerations of the human factors. The design of the man machine interface for computer systems is still in its infancy - still an art rather than a science - but certain devices and procedures can be used to build user friendly systems and these are discussed in the following sections.

3.3.1 Interactive Devices

The interactive device provides a means of conversing with the computer. The traditional input device is the alphanumeric keyboard, requiring a certain amount of typing proficiency. There are however, a number of interactive devices becoming increasingly popular, both with system designers and with system users.

3.3.1.1 Logical Devices

A logical device is used to perform a specific interactive function. These can be separated into locator, pick, valuator, keyboard and button.

The 'locator' is a device that enables an absolute position, or direction, to be input to the system. These are the most common form of interactive device as they can be used to simulate many other logical devices. The tablet with a hand held cursor or stylus, the mouse, the trackerball, the joystick and the touch panel are all forms of locators.

The most used locator device is the mouse. This can either be mechanically or optically driven. Mechanical mice use potentiometers to measure the movement of a ball held in a socket. This ball rests on a surface, moving the mouse makes the ball roll, the movement of the ball is measured and its absolute position or direction determined. Optical mice generally require the use of a pad marked with a coloured grid. The mouse emits different colour lights from two sources, this light is reflected off the pad into detectors which pick up the movement across the grid. Nearly all locator devices also have a number of buttons. These buttons can be used to indicate features other than direction.

The 'pick' is a device that is used to select a displayed option. That is to point at a particular entity and 'pick' it. The only true 'pick' is a light pen connected directly to the Display Processor.

The 'valuator' is a device that inputs a number from a range: the logical device for this is a potentiometer dial.

In a truly interactive system keyboards are used only to input character strings. Buttons are similar and can be part of the keyboard as a programmable function keypad. Each function key selects an entity.

3.3.1.2 Simulation

In reality very few of these devices find their way into everyday use. The normal procedure is to use one or two interactive devices. The remaining logical devices are then produced by simulation.

The 'locator' can be simulated in a number of ways. Perhaps the most common is a cross hair cursor driven be a stylus, mouse or even particular function keys. The lowest form of locator interaction would be entering the position using a keyboard.

The 'pick' is also easily simulated using the above devices. Buttons on the hand held cursor are then used to select the entity. The use of multiple buttons is particularly effective with pick devices. A further simulated pick device is the intensification of options in a cyclic fashion. That is each option is highlighted in a cycle on successive presses of a button until a selection button is pressed.

The 'valuator' can be simulated in two ways. The first is entering the value using the keyboard. The second method displays the range of available values on a scale or dial which is then altered until the required value is indicated.

3.3.1.3 Menus

The use of menus is a common button simulation. Menus display the possible alternatives in the form of text strings or icons, which are a pictorial representation of the option. Option are selected using function keys, key letters or some form of 'pick' device.

Menus are an important interactive device because they minimise the required memorisation of the meanings of the options. Command names are replaced by recognizable icons or descriptions. In a command driven system the user must have full knowledge of all commands and the parameters they require.

There are a number of rules to adhere to with the compilation and presentation of menu systems. A screen full of wordy options can be difficult to use because the visual context is lost. Each menu option should therefore be abbreviated as much as is possible without losing recognition of the function. In addition it is advantageous to group classes of options. This keeps the number of options in each menu to an easily delineated set and forces the system to a hierarchical tree structure. Experience with the use of menus, designed for the water information system, suggests that the number of options on each menu should be kept to below 10, preferably 5 or 6.

The design of a hierarchical menu structures should also consider the nesting of the menus or groups of functions. Menus are said to be nested when selection of an option produces a further menu. This nesting continues until the routine is accessed. It is advisable to keep the level of nesting low. To be forced to travel through more than 2 or 3 menus to reach a function is excessive.

The positioning of the menu on the screen is also important. Ideally the menu should be placed where the function is performing, the user's centre of concentration. This however can have adverse effects due to the alteration and temporary loss of the image. This form of menu - that appears when required and then restores the display - is known as a dynamic or pop-up menu. Static menus are menus that are always present, at the top or side of the display. Using static menus requires a shift in the centre of concentration, from the image to the menu area.

3.3.2 User - Computer Dialogue

User - computer dialogue is like any other language; it has rules of grammar and a vocabulary, and it involves a learning process. Perhaps the most fundamental rule in the design of dialogue is to keep the rules of grammar and vocabulary simple. The dialogue should use known concepts, or introduce only easily learnt concepts. To a large extent the dialogue should be performed on a 'need to know' basis, and should not be over-complicated with lower level function calls, such as graphics primitives.

In any conversation a question is asked and an answer is expected. If this answer is not immediately forthcoming then some sign of attentiveness is required until the answer is available. This attentiveness is known as dialogue feedback. There are therefore two languages, that of user to computer and of computer to user. Both languages are described in terms of their conceptual, semantic, syntactic and lexical design.

Conceptual design is the definition of key concepts which the user must be aware of and have mastered. Semantic design is definition of the functionality and information needed, and description of the errors that could occur. Syntactic design describes the sequences of input/output tokens that make up the language, and defines the grammar. Lexical design determines the formation of the input/output tokens from hardware primitives, for example - keyboard characters.

3.3.3 Interface Design

The design of an interactive user interface is largely an art and as such has few rules. A number of principles can however be suggested.

1. Knowledge of the functionality of the system.
2. Minimisation of the memorisation required.
3. Robustness of the system.

One of the most important principles in design of an interactive interface is linked to the requirement specification stage of the software engineering process. That is to know the user of the product and the tasks that will be performed on the system in its operating environment.

With large amounts of information appearing on the screen certain areas may need to be masked. This requires the user to memorise information, a further design principle is to minimise the memorisation required in operating the system. Each operation should then be optimised to provide minimum screen clutter, displaying only useful or required information when requested.

In addition the system must be robust, in that all user errors must be trapped by the system and explained. Failure to do so can result in frustration and confusion. Boredom, due to improper response times; panic, at lengthy delays in waiting for a response; frustration, at not being able to get the required action understood; and confusion, in not being able to differentiate between available options; should all be accounted for. Failure to do so could result in the user losing confidence in the system.

These design principles can be categorised in terms of feedback, learning, management of errors, response time and the display.

3.3.3.1 Feedback

Feedback is computer to user dialogue, that is the response given to a particular user input. A system with no feedback is analogous to talking to someone who does not acknowledge the presence of the speaker - generally a frustrating experience! The normal mode of operation for computers is one of no feedback: the exception to this is the 'power on' light. Provision of feedback becomes the task of the programmer. In terms of the components of a dialogue, feedback can be classified as lexical, syntactic or semantic.

Lexical feedback is the lowest form of feedback, and can simply consist of echoing characters typed on the keyboard on the display or a screen cursor moving as the mouse is moved.

Syntactic feedback consists of dialogue that tells the user a command has been accepted and understood. This can take the form of prompting for the next input or the operation of the selected function.

Semantic feedback is perhaps the most useful as it tells the user that the task has been completed. The use of counters informing the user on the progress of a process, and the display of partial results are further forms of semantic feedback.

The location of the feedback on the screen is an important factor, and should not move the user's centre of concentration. Fixed areas for messages can be distracting for this reason. Ideally the feedback should be placed in the area of interest, for example at the current cursor position.

3.3.3.2 Learning

The system should be designed to cater for all types of user from the novice to the experienced.

Perhaps the most useful aid to the novice is a guided tour of the system under system control. That is the user sits and watches whilst a typical interactive session is simulated. Further aids for the novice are prompting and suggestions as to the required input, the use of menu systems and the provision of an on-line help facility, available at any point in the session. On-line help facility does however require a certain amount of memorisation, as it generally replaces the current display.

Conversely, the experienced user has to be able to access the functions of the system in rapid succession. Menu systems are suitable for this type of user only if the menus are presented quickly. Perhaps more useful, if the function name and arguments are known, is a command line interpreter. This enables fast access of functions throughout the system. Command line systems invariably allow frequently used sequences of functions to be stored in a macro command file. This macro can then be invoked by the user on the command line.

3.3.3.3 Error Management

The ability to correct or rectify an error is fundamental to any interactive system. Accommodation of errors and provision of backup is therefore important in reducing frustration and increasing productivity. The provision of adequate error checking can speed up the learning process and free the user from 'fear of failure'. The simplest method of accommodating errors is to keep a system of backups and allow the process to be aborted. The user is then returned to the previous position and allowed to continue. This generally requires storing multiple versions of the screen on file, which with large scenes may prove impossible. A further method is to allow the replaying of all user input from a stored file up to the point where the error occurred.

With some processes, for example deleting files, explicit confirmation of intention and the opportunity to cancel the command can be used.

3.3.3.4 Response Time

The response time of a system depends on the question asked and the machine used. This is also related to the expectations of the user.

For example there are certain functions where responses should be immediate, such as typing a character and moving a cursor. For more involved processes the response time may be slightly longer. It has been suggested that the maximum time a user can wait for a response to a more involved process is about 2 seconds. The expected response time is related to the user's knowledge of current computer technology. This expected response time should be the limit for feedback intervals.

3.3.3.5 Display

The general rule for organisation of a display is that it should appear uncluttered. This can be achieved by limiting the number of recognisable elements and organising the screen effectively, for example by the use of borders.

The perception of the structure of the items should therefore be enhanced with visual encoding such as colour, line-style and intensity. Of these the easiest to perceive is colour. The maximum number of codes for near error free recognition of elements for different coding methods is presented in table 3.1

Coding Method	Maximum Number of Codes
Colour	6
Shape	10
Line Width	2
Line Type	5
Intensity	2

Table 3.1. Maximum number of different coding methods for near error free recognition, Foley and Van Dam (1984)

Any of these codings can be combined to aid discrimination and increase the maximum number of recognisable elements.

3.4 Image Processing Technology

Over the last few years there has been a vast increase in the power and availability of image processing products. The advent of board level expansion products has meant that professional systems can now be developed on micro-computers. In the field of remote sensing this is demonstrated by the increasing number of IBM PC based image processing systems currently available.

The majority of these products have been developed by manufacturers of the larger mini- and main-frame hosted image processing systems and consequently have adapted preexisting software, rather than develop new software using the features of the boards. In some cases the board is seen as little more than a local, limited resolution, display device.

In addition to use in an image processing system or workstation there are a number of further applications of this form of product. For example, the board generally provides large amounts of video memory. This memory has two ports, one port is in a continual read cycle supplying the display processor and screen with data; the other is available for read/write operations. This means that data can be written to the memory and simultaneously displayed. The advantage of this is that results are immediately apparent and errors easily found. The memory can also be thought of as a large two-dimensional array, and can therefore be used as input and output in, for example, interpolations and mathematical models.

The technology of these products has been tested for a number of applications. Chapter 6 details application in interpolation routines. Prior to this it is worth describing the relevant features of these products.

3.4.1 Image Processing vs Graphics Boards

For the purposes of this text the term image processing boards is used to describe a subset of graphics boards otherwise known as raster scan display boards. The difference between these boards and random scan display boards is in the display processing unit.

Figure 3.2 shows the layout of a typical computer graphics system. The main four components can be identified; the host computer, display processing unit (DPU), display and the user interface. Common peripherals, the printer and plotter, are also shown. This layout is common to both types of display processor products.

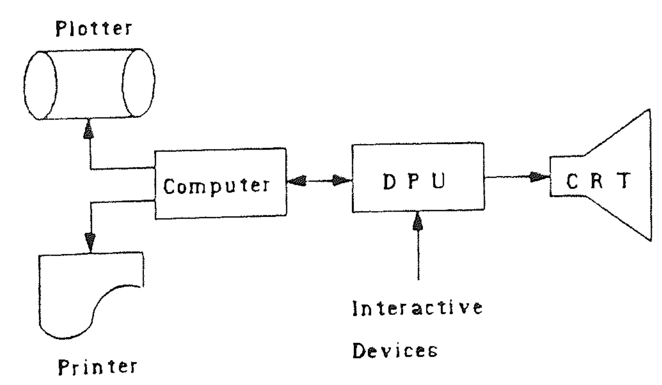


Figure 3.2. The Computer Graphics System.

3.4.1.1 Random Scan DPUs

The Random Scan DPU is the heart of the vector graphics board. This DPU is designed to draw points and lines from vector lists at high speed. Figure 3.3 is a schematic of this form of DPU.

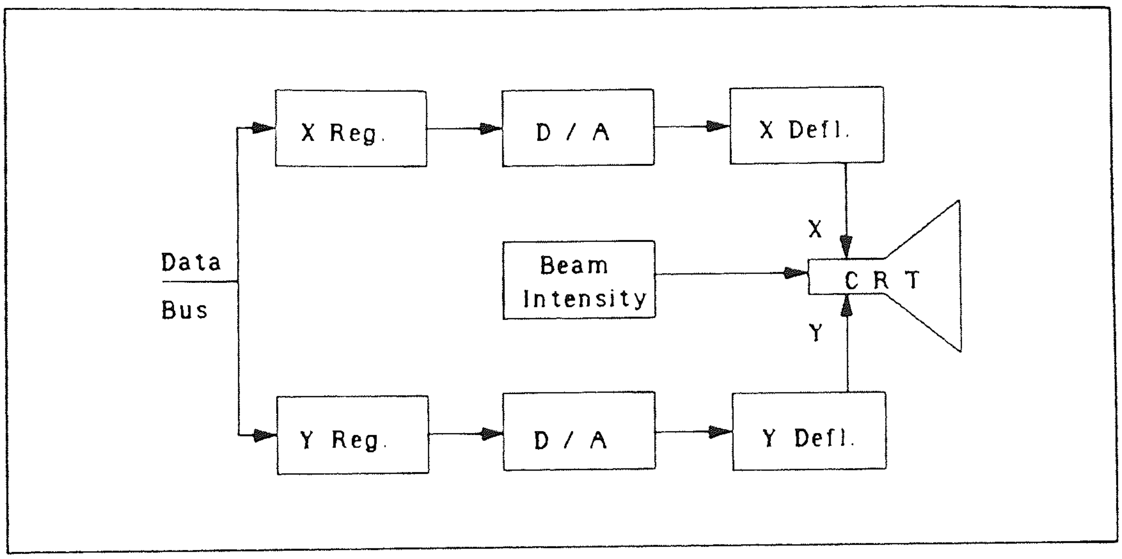


Figure 3.3. The Random Scan DPU.

The memory on board, known as the display list memory, holds a list of horizontal and vertical co-ordinates. This list is accessed via the bus and loaded into two registers point by point. The registers are then converted to corresponding beam deflections which drive the magnets on the cathode ray tube. In addition the DPU controls the intensity of the beams hence altering the colour displayed. Each point stored in the vector list is accessed and the beam moved. Thus the beam can move at random to any point on the display: hence random scan DPU.

The whole process is repeated in a cycle that maintains the persistence of the phosphor on the display. The repeat cycle is known as the refresh of the screen. The rate of refresh of the screen for a random scan DPU is therefore controlled by the complexity of the image.

This is the simplest form of DPU. More complex DPUs take a large amount of the processing away from the host. As an example the drawing of a line would normally require the host to calculate the position of every relevant point on the line and store these points in the vector list: the DPU is usually programmed to perform a number of these basic functions and therefore limits the processing required on the host.

3.4.1.2 Raster Scan DPUs

The organisation of the Raster Scan DPU differs substantially from the Random Scan DPU. Figure 3.4 is a schematic of the Raster Scan DPU.

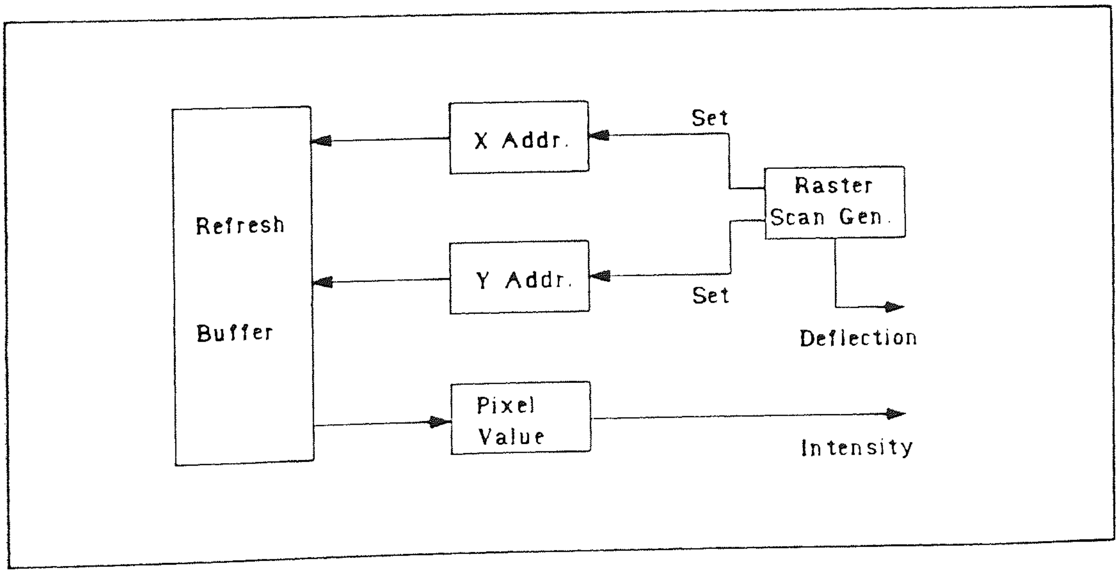


Figure 3.4. The Raster Scan DPU.

The major difference is the organisation of the memory. This memory is arranged as a two-dimensional array of picture elements (pixels), known as a refresh buffer or framestore. Each graphics feature is then written to the corresponding memory location. The DPU then scans the memory locations sequentially in the horizontal direction. The memory address of each pixel is determined by a raster scan generator. Each address is then accessed to determine the brightness and colour of each pixel. The electron beam deflection is then controlled by the raster scan generator, the intensity controlled by the memory address value.

The memory used in raster scan systems is known as video RAM. This memory has two data ports allowing simultaneous access of the memory by the raster scan processor and the host processor.

Because every pixel in the framestore is displayed, a faster refresh is required. For this reason some raster scan generators split the image or frame into two interlaced fields, consisting of the odd or even lines of the image. Each field is refreshed every 1/50th second, the total display is refreshed in 1/25th second. It is this interlacing of fields that causes flicker on the display, especially on displays with short persistence monitors. Due to the amount of data having to be scanned the present limits for available interlaced systems is about 768 x 576 pixels.

Non-interlaced refresh requires a faster scan rate, and is consequently more expensive to implement. The required data rates become prohibitive when high resolution (1280 x 1024 pixels), full colour (24 bits/pixel) displays are required, and consequently many high resolution displays have a limited colour palette.

3.4.1.3 Hybrid Products

In addition to random and raster scan DPUs, a number of computer graphics products utilise features of both products. These products can therefore be used for both vector and raster graphics.

A typical example of the hybrid graphics system is represented in Figure 3.5. This system is essentially a raster scan system with a display list processor simulating a random scan DPU. This format is becoming increasingly common in vector graphics workstations and has the added advantage of allowing programmable colour palettes through the use of Look up Tables.

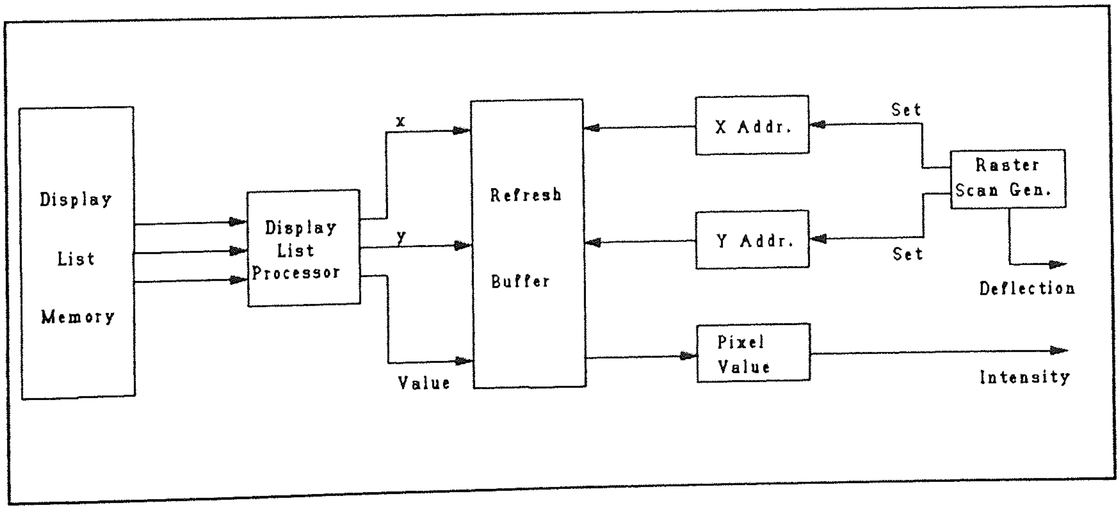


Figure 3.5. The Hybrid Graphics System.

The display list processor is generally a conventional programmable processor such as the Motorola 68020 system detailed in 3.4.3.2, or can be more powerful processors such as transputers or digital signal processors (DSPs).

3.4.2 Components of the Graphics Board

For image processing of remotely sensed imagery and general manipulation of spatial data, graphics systems using framestores are distinctly advantageous. Raster display can be performed on vector graphics systems, but the utility of such systems is rarely more than display, formed by simulating a raster scan with adjacent colour filled rectangular blocks.

For both raster and hybrid graphics systems there are a number of common components. The components that typify a system for image processing specifically are the Framestore and Look up Tables.

3.4.2.1 Framestore

The framestore is an area of fast switching or dual ported video memory which acts as an array of read/write locations. This memory can be read or written to, and is simultaneously displayed on the output screen by the display processor. Each memory location is a screen pixel (picture element). The memory is usually a minimum of 512 x 512 pixels and is presently limited in micro-computer applications to about 2048 x 1024. Each location can be 8, 12, 16, 24, 28 or 32 bit memory. This accounts for 8 bit/pixel images, 8 bit/pixel images with 4 bit overlays, 12 bit/pixel images, 16 bit/pixel images, 24 bit/pixel images and 24 bit/pixel images with 4 or 8 bit overlays.

Of these, the most common are the 8 bit/pixel and 24 bit/pixel memory with 2, 4 or 8 bits of overlay. These two options are reflected by the two boards studied in this chapter.

3.4.2.2 Look Up Tables

Look up Tables are a standard method of performing fast manipulation of large amounts of data. If the range of data is limited, for example an 8 bit variable with a range of values from 0-255, then Look up Tables, or LUTs, provide the fastest possible method of performing an operation. Consider the action of multiplying 1024 8 bit values by 0.5. This in conventional terms would mean performing 1024 multiplications. If a table were prepared containing the entire range of possible input values, 0-255, and for each input value the output(the value multiplied by 0.5) the process would then involve only 256 multiplications. The saving in operations is 1/4. If however we multiply 1048576 values (1MB) the saving is 1/4096.

The LUT can be seen as an array of memory locations where the input is the offset into the array and the output is the value stored at that offset.

On image processing boards the LUTs are normally a separate portion of memory accessible via the data bus. The display processor accesses the framestore through these LUTs and therefore altering 256 values in the LUT can alter the entire display of the framestore in the time it takes to multiply and write 256 values. This time is generally less than 1/25th second ,the rate of refresh; the function appears to perform instantaneously.

The majority of boards have at least three LUTs, one for each of the Red, Green and Blue outputs of the system. In an 8 bit system the three LUTs can be altered to give different amounts of Red, Green and Blue output for a single input. Thus grey scale images can be made to appear coloured. This process is known as pseudocolour operation, and in particular colouring ranges of memory values is known as colour or grey scale density slicing.

The major advantage of Look up Tables in image processing is in the alteration of the image contrast and brightness. Using LUTs the display can be changed without altering the pixel values stored in memory. Thus contrast and brightness can be altered interactively to produce a visually enhanced image with the knowledge that the original pixel values are unchanged, with no loss of accuracy.

3.4.2.3 Accelerator Board Products

There are a number of additional processors and board products that can be used to boost performance of the image processing graphics board. To be usable, the architecture of the hardware has to be open, with transparent, preferably direct, access to the framestore.

The kind of operations these boards perform are generally designed for applications in industrial image processing and robot vision where real time processes are required. Some lend themselves to all applications for example, general ALUs (Arithmetic Logic Units).

Examples of additional functions are hardware histograms, fourier analysis, geometric manipulations and parallel processors for operations such as convolution filters.

These boards are normally supplied with basic software and to fully utilise these products it is necessary to write low-level code. It is not always easy to integrate these products, the investment of time and effort in producing a workable system increases considerably.

3.4.3 Selected Graphics Products

A number of graphics products were evaluated for processing remotely sensed data, input of cartographic data and manipulation of general spatial data. Two board level graphics products were investigated, Imaging Technology Incorporated - PCVision+ and Real World Graphics - PC4000. Both boards are IBM PC-AT compatible expansion products. The boards both have framestores, Look up Tables and support video digitisation. The boards differ in that the PCVision+ board relies on the host processor, the PC4000 has an on-board Motorola 68020 with 1MB RAM for processing. These products are typical of a raster scan graphics and a hybrid graphics system.

3.4.3.1 ITI PCVision+

The board studied was supplied by Amplicon Electronics Ltd, who are the distributers for Imaging Technology Incorporated, USA. This company specialises in image processing products for the industrial and medical markets. The boards are therefore not specifically designed for normal 24 bit colour remote sensing operations. However 3 PCVision+ boards can be configured to provide 24 bit operation.

3.4.3.1.1 Description

Figure 3.6 is a schematic diagram of the PCVision+ Board. For more details refer to the PCVision+ Users Manual, Imaging Technology Inc. (1987).

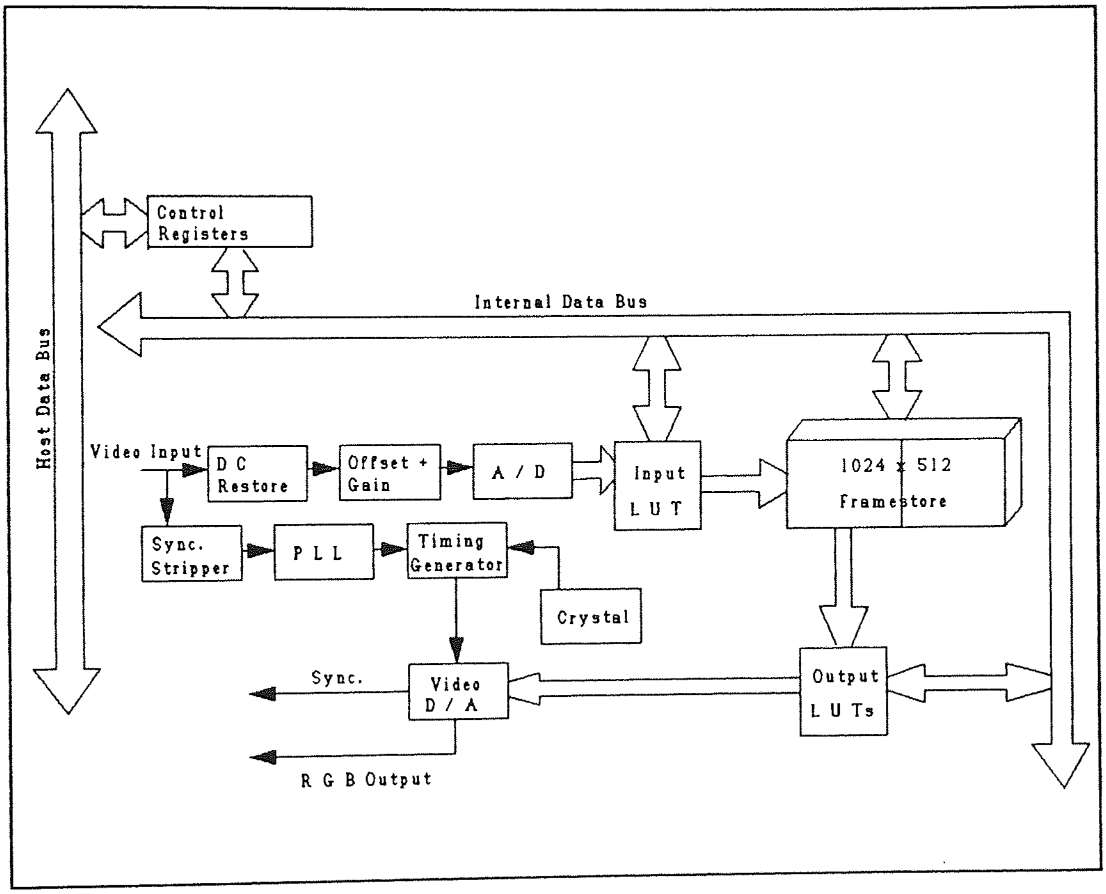


Figure 3.6. The ITI PcVision+.

The framestore is 1024 x 512 pixels configured as two 512 x 512 pages of dual ported video memory. The display processor allows a 2 x zoom and panning and scrolling in 8 pixel jumps. The output from the board is a 512 x 486 pixel window on the memory which could be altered using the zoom, pan and scroll.

In addition to the framestore the board had 8 banks of Look up Tables. Each bank consists of four LUTs; one for the red, green and blue outputs and one for the input.

The input LUTs are present to allow alteration of a video input signal prior to digitisation and writing to the framestore. The video digitisation option is common to boards designed for industrial and medical image processing, and allows a composite video signal to be framegrabbed or digitised. This has further application in digital mapping and remote sensing which is more fully described in Chapter 4.

The framestore and LUTs are all accessed via an internal data bus which is controlled by registers. These registers are controlled via the host expansion bus by the host processor. The board's registers are then accessed by the host at a particular port address.

3.4.3.1.2 Operation

The host PC has a large area of unmapped allocated memory addresses. That is, particular memory locations can be accessed above the RAM (Random Access Memory) of the computer. This is made use of by, for example, the graphics or text display adaptor of the host machine. The PCVision+ board works in a similar way. A memory segment of 64KB is mapped at any one time into the host memory space. The entire framestore on the board is split into 8 blocks, each being 512 x 128 pixels (64KB). One of the registers on the board, accessed on the port, controls which of the blocks is memory mapped at any one time. Once the block is memory mapped, reading or writing pixels is simply a case of determining the offset in pixels from the start of the block and reading or writing from the corresponding memory location of the hosts memory space. Communications are thus relatively simple.

In addition, with 24 bit systems each of the 8 bit boards must then sit at a unique port and memory address to prevent conflict.

3.4.3.1.3 Read/Write Pixel Functions

The routine to read a 24 bit pixel, perform an operation on the three 8 bit components and write the result back to the same location can now be described.

For each board:
- Set Port and Memory Addresses
- Initialise Board
- Determine which 512 x 128 block
- Map block into memory
- Read memory location
- Disable Board

Perform operation on red, green and blue pixels:

For each board:
- Set Port and Memory Addresses
- Initialise Board
- Determine which 512 x 128 block
- Map block into memory
- Write memory location
- Disable Board

Therefore it can be seen that a large proportion of the code is the switching on and off of the relevant boards and memory blocks.

3.4.3.2 RWG PC4000

The board studied was supplied by Real World Graphics Ltd, a company which specialise in imaging and graphics hardware for the CAD and computer graphics market. There are two distinct lines of products, high resolution, high speed vector graphics systems based on transputers and RISC processors, and lower resolution full colour products designed for video and television graphics. The PC4000 can be described as a hybrid graphics product, with video digitisation, framestores, Look up Tables and an on-board Motorola 68020 for display list processing.

3.4.3.2.1 Description

The board has a 768 x 768 pixel framestore each pixel being 28 bits deep. This can either be used for full 28 bit pixel operations or using three 8 bit and one 4 bit component. The latter is most suited to remote sensing applications. The memory is 32 bit video RAM, arranged with the bottom eight bits displayed as red, the next eight as green, the next eight as blue and the bottom four bits of the top 8 bits displayed as overlay planes. The memory is not dual ported, instead the memory is accessed by the on-board processor and raster scan DPU in rapid succession via a switch.

Figure 3.7 is a schematic diagram of the Real World Graphics board. For more details refer to the PC4000 Users Manual, Real World Graphics (1988).

The output from the board is a 768 x 576 window on the memory, which is essentially the standard TV display. This output can be altered using a 1 to 16 x zoom and pan and scroll in single pixel increments. The output is also controlled by three Look up Tables, red, green and blue. These LUTs have a 12 bit address with 8 bit output, enabling the 8 bit display components to be altered and to use a 16 colour palette for the 4 bit overlay plane selectable from a palette of 16.77 million.

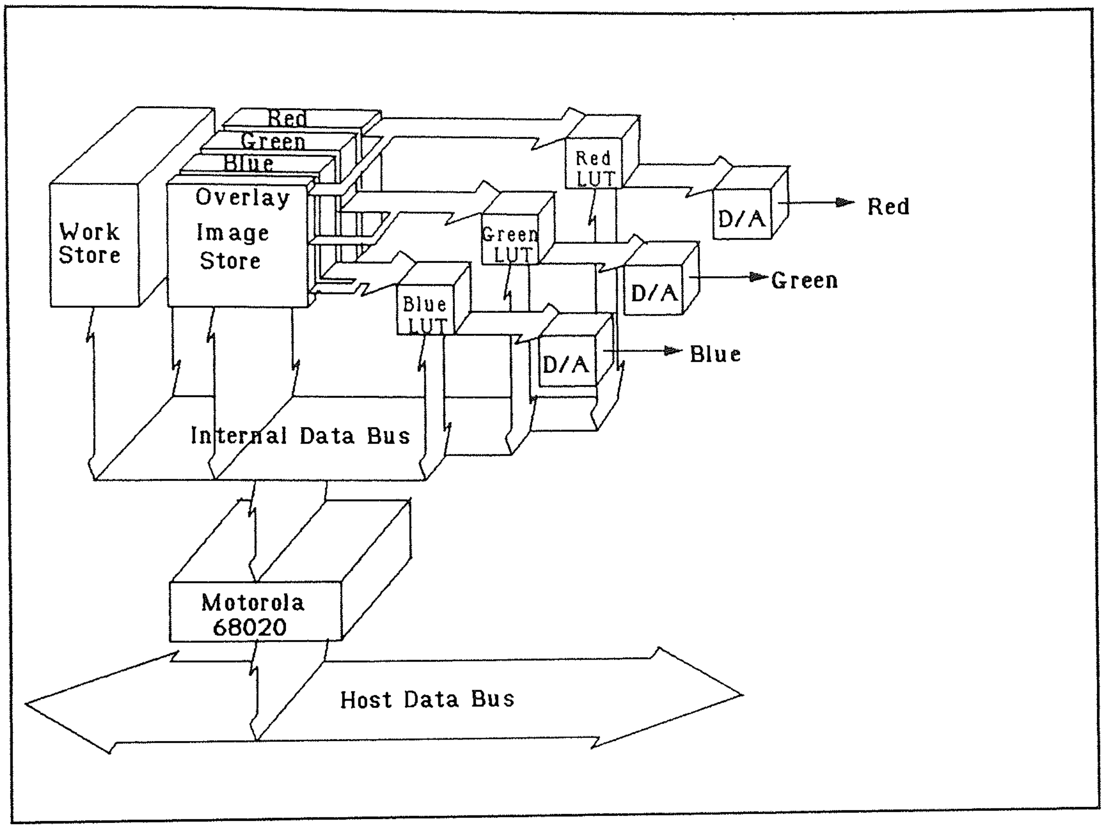


Figure 3.7. The Real World Graphics PC4000.

The processor on board is a Motorola 68020 operating at 16.67 MHz. This processor provides fast 32 bit integer arithmetic and can access the framestore and 1 MB of user memory via an internal 32 bit data bus.

The overall graphics system is therefore essentially a two processor system with the on-board processor handling the image processing operations and the host processor providing the user interface. Communications between the two processors is via a port and has a controlling protocol.

3.4.3.2.2 Operation

The major features of the operation of the system are the communications protocol. On booting up the system an EPROM on the board takes control and runs a server routine. This server enables the image processing functions to be performed under the control of a series of codes passed to the server by the program running on the host.

The first stage in the operation is to initialise the board and load some firmware routines into the workspace of the 68020 processor. This firmware contains a number of simple routines, which are invoked by passing a control code to the server. The host sends a series of control codes to the port address where the image board is located. The host must take into account the necessary communications protocols, and wait for return values. In addition to this form of operation, code can be produced on the host, cross compiled and assembled into Motorola S-Records and down loaded into the workspace to perform operations. The down loaded code is then invoked by sending control codes as before.

Since the video memory is mapped into the memory space of the 68020, the routines access pixels using direct memory addresses. The pixel position being a four byte address, the colour components a byte offset from the address. The 68020 without a math co-processor was found to perform slower than a 16.67 MHz host processor for floating point operations. Floating point operations were therefore shared between the two processors. Inherent floating point operations, for example maths functions such as sine,logarithms and square roots, were performed in the host. Floating point operations such as multiplication and division were performed on the 68020 by coding floating point values into scaled fixed point (integer) values.

The communications link with the host proved to be extremely slow when reading or writing small areas. This was because of the handshaking required for each byte transferred. The majority of operations performed on the host therefore worked on lines or areas of data to cut the communications handshaking to a minimum.

3.4.3.2.3 Read/Write Pixel Function

The routine to read a 24 bit pixel, perform an operation and write the transformed pixel back is performed as follows.

Host (Software)	Board (Firmware)
Send Code to Read Pixel	Accept Code
Send x,y position	Accept x,y
	Determine memory offset
	Read memory location
Receive 24 bit value	Return 24 bit Value
Unpack into 8 bits	
Perform Operation	
Pack to 24 bit	
Send Code to Write Pixel	Accept Code
Send x,y position and value	Accept x,y,value
	Determine memory offset
	Write memory location

The sending and receiving of codes and values, or handshaking, slows the process considerably. For this reason a line of pixels is normally read, returning an array of values, handshaking occurs only at the start and end of the line. For these operations the sending of a line is almost as fast as the sending of a pixel.

3.4.4 Applications

There are numerous applications of imaging technology, and in particular spatial manipulation of data stored in framestores. A number of these applications are detailed in the following chapters on video digitisation, interpolation and airborne video remote sensing. These include the straightforward processing of airborne and satellite remotely sensed images to produce thematic products; the production of raster and vector digital databases of features, extracted from maps and charts using video digitisation; and the use of imaging technology to store image and map data in a format facilitating exploitation of spatial relationships.

3.5 Data Communication

For a system to be fully integrated into the environment it must have communications linking it to networks of workstations or computers and data collection points.

Communications with networks of computers is now fairly standard. There are numerous networking systems available, enabling access to data throughout the system by any component of the network. Networking systems can also handle the necessary communications protocols enabling machines of different internal architectures to communicate.

Automated collection of environmental data is now possible through the use of Data Collection Platforms (DCPs). These platforms make use of satellite telemetry to transmit hydrological measurements from standard sensors to receiving systems.

In addition Secondary Data Users Systems (SDUS), capable of receiving DCP data as well as Meteosat image data, can be obtained for very little cost. More expensive Primary Data User Systems (PDUS) offer the advantage of higher resolution and coverage. At a higher resolution, both spatially and spectrally, the NOAA-TIROS series offer twice daily coverage with a whole range of sensors. With data from DCPs to calibrate the image data, its full potential for water resources applications can be realised. These are what are known as real time communications. However the acquisition of high resolution satellite data is achieved via expensive tracking ground receiving systems based throughout the world. Dissemination of these kind of satellite data is achieved in the space of a few weeks, and is generally expensive.

3.5.1 Data Collection Platforms

The late 1980's represents the beginning of the Data Collection Platform explosion. Within a few years a massive network of DCPs will cover the globe, as the meteorologists, hydrologists and environmentalists become aware of its potential. Already there are numerous systems installed measuring rainfall, temperature, humidity, soil moisture, water level, channel flow, wave height etc., throughout the world on marine buoys, automatic weather stations and hydrological monitoring stations. An important feature of these data is that any transmission using the satellite communication channels can be accessed and decoded if the structure and format is known.

The DCP consists of a radio transmitter containing a data acquisition and processing unit. This is connected to a set of sensors recording any of a number of environmental factors. The DCP operates on a self timer that transmits the data stored at an allocated time. Alternatively DCPs can be triggered to transmit by a specific event, this is of particular use in disaster monitoring.

Both geostationary and polar orbiting satellites can be used to relay data. Geostationary satellites, such as Meteosat, are more widely used due to the synoptic, continuous coverage they provide. With about 60 radio channels the satellite can relay up to 40,000 message per day.

To receive the information from the DCPs via the satellite a ground receiving station is required. The Meteosat Control Station, near Darmstadt in West Germany, can be used with dissemination via telex. Alternatively a satellite receiver-processor system can be used. This system includes a fixed position 1.5m dish antennae connected the receiver unit. This can be interfaced to a micro-computer, such that an archive of data can be updated on a regular interval and stored on the computer for later processing.

3.5.2 Meteorological Satellites

There are a number of meteorological satellite systems which can be received within an organisation such as a statutory body. These are the Meteosat and NOAA-TIROS series. The cost of the receiving systems for other satellite systems prohibits their use for real time acquisition.

Meteosat is one of a global network of geostationary satellites located at an altitude of 36,000 km above the equator at 70° longitude intervals. At present Meteosat 3 is in operation demonstrating the European Space Agency's aim of maintaining a geostationary satellite in orbit. Image data is acquired in three wavebands: Visible, Thermal Infrared and Water Vapour Absorption. The resolution of the sensors is 2.5 km in the Visible and 5.0 km in the Infrared at the nadir (0° Lat., 0° Long.). The satellite acquires an image over the entire earth disc every 30 minutes.

The data can be received in two formats. Using a Secondary Data User System (SDUS) images can be received in an analogue low resolution weather facsimile mode (WEFAX). Alternatively a Primary Data User System (PDUS) can receive the data in high resolution mode. High resolution data is required for most hydrological applications.

The NOAA-TIROS series of satellites are in a sun-synchronous near polar orbit. At present there are two satellites viewing both day and night sides of the Earth for two global views each day. At an altitude of 833-870 km the satellites cross the equator at 0730, 1930, 1400 and 0200 local sun time. Thus two separate views of a particular area can be obtained each day. There are two main sets of instruments on-board, the Advanced Very High Resolution Radiometer and the TIROS Operational Vertical Sounder. The AVHRR has 5 channels of image data covering the visible to thermal Infrared wavelengths at a resolution of 1.1 km. The TOVS has a number of sounding units giving profiles through the atmosphere at varying resolutions.

Data can be acquired through one of the national ground receiving stations, alternatively a receiving system with a tracking antennae enables real time data acquisition of the data. These systems are still expensive, and it may be more viable to obtain this data through ground stations.

4 CARTOGRAPHIC DATA INTEGRATION

4.1 Introduction

The input of cartographic thematic, topographic and other data is the necessary first stage in setting up a spatial database for a project area. The production of the database lies at the core of the entire system and is frequently the most time consuming and expensive stage of a study. It is therefore sensible to investigate methods of input of cartographic data in terms of reliability, accuracy, speed, ease of use and cost.

Possible input methods include the use of bit pad digitisers, drum scanning densitometers, document scanners and video digitisation. In all these methods there is a need for human interaction, as at some stage each feature has to be manually followed with a cursor.

Alternative innovative methods come under a certain suspicion. This is mainly due to the high cost of map digitisation, as a percentage of the total database production. However these methods are being introduced, and many of the larger mapping companies now market products and services in map digitisation using automatic methods; a notable example being Laserscan Laboratories Ltd.

Automatic and semi-automatic methods of digitising feature data have been proposed and a limited semi-automatic system for line digitisation was developed. However it was decided that these algorithms were not suitable for general feature extraction, until the intelligence of human spatial conception could be incorporated. Manual methods initially provide higher accuracy and reliability.

4.2 Bit Pad Digitisers

This is the recognised standard for input of cartographic data. Digitising tablets come in range of sizes from A4 up to A0 size. With all systems the tablet consists of a sensitised grid which picks up the position of the cursor to a high degree of precision. The cursor is a hand held instrument with a fine cross hair and number of buttons controlling 'pen-up' and 'pen-down' functions.

The interface to the host machine is generally a port on the host data bus. Data is communicated to the host which allows the x,y co-ordinates of the vector to be derived.

The normal mode of operation is to first initialise and characterise the feature. Then the cursor is moved to the start of the feature and the start point marked with a button press. The cursor is now 'live'. Once set the current position will be output every time the cursor is moved a set distance or when a certain amount of time has elapsed. The feature end point is marked by a second button press.

The digitised accuracy of the feature derived then depends on the scale of map used, and more importantly how closely the feature is followed. The possibility of errors occurring can be quite large due to cursor slip, confusion and repetition.

This is the method preferred by the majority of mapping organisations and the Ordnance Survey employ this method to transfer the vast catalogue of OS cartographic data to digital format.

4.3 Drum Scanning Densitometers

Drum scanning densitometers provide a method of producing a raster representation of the map. To do this the map is fixed to a drum which is rotated at high speed. During rotation an arrangement of lasers and sensors illuminates the map and measures the reflectance in the primary colours. This can be done to a very high level of spatial resolution, in the order of microns, though for map data this accuracy is not generally required.

This method is similar to that of video digitisation described in the following sections. The drum scanned images are generally of very high radiometric quality and potentially can be used in automatic feature vectorisation and for providing a raster representation of a map as a background to superimpose digital data. It is however an expensive method of input; drum scanning densitometers costing in the order of £35,000. There are however companies that provide this as a service, costing about £100 per A4 sheet.

It is interesting to note that the Ordnance Survey contracted out digitisation of contours from topographic maps and subsequent production of digital elevation models. For this purpose the Ordnance Survey issued a map marked only with contours, which was subsequently scanned. Automatic feature extraction, or 'line following' as it is often known, is relatively simple with uncomplicated features that are complete and do not cross (not always the case with mapped contours!). It is extremely difficult to extract contours, even manually, from a complete topographic map. For this reason the automatic extraction of features was attempted only with single feature maps. The alternative is to manually trace off the contours, then digitise the tracing, which amounts to more work than manually digitising the contours.

4.4 Document Scanners

The recent growth in the desktop publishing market calls for a method of integrating hand drawn diagrams and photographs into text. Generally these images are used as monochrome images, with the various tones being produced by dithering dot patterns or halftoning. The majority of scanners consist of a flat bed platen glass, on which the image is placed. The image is then illuminated and scanned by a moving array of sensors. Depending on the hardware configuration and available resolution of scanning the systems often require extended host memory and large fixed disk drives. A number of 'standards' for input and storage of scanned image data have also rapidly evolved. These include PICT, Encapsulated Postscript and TIFF (Tag Image File Format).

In addition to the flat bed scanners, sheet fed and hand held scanners are also available. Sheet fed scanners work on a similar principal to line printers, with the image being fed under a stationary array of sensors. Hand held scanners rely on the operator to scan an image. Both sheet fed and hand held scanners have problems of geometric distortion.

The low cost document scanners now available can digitise A4 sheets to 1,2,4,or 6 bit precision, that is up to 64 grey levels, at resolutions from 75 to 300 dots per inch (dpi). More expensive scanners can handle up to A0 size drawings at resolutions up to 800 dpi. Digitisation using desktop scanning devices appears to be midway between video digitisation and drum scanning in terms of spatial and spectral resolution, and may well provide a useful compromise.

Using such a system map sheets would be divided into A4 sectors and digitised either in colour or black and white. Colour scanners are currently expensive, colour digitisation can however be achieved, if a white light source is used in the scanner, by using sheets of coloured acetate.

One of the major advantage of the flat bed document scanner is that the chart is placed on a flat platen glass and is illuminated by a strip light at the same time as the scan. This ensures constant illumination across the chart. In the absence of vignetting and shading automatic line following is considerably simplified.

4.5 Video Digitisation

Video Digitisation is a process which enables any image viewed by a video camera to be digitised. That is, the brightness of the pixels on the image are converted from analogue signal amplitude to a digital number.

The image can then be represented by an array of numbers, the higher the number, the brighter the pixel on the image.

This is the format handled by computer graphics and image processing systems, and therefore any image viewed with a video camera can be digitally enhanced and processed.

Using this readily available technology, video cameras can be mounted in planes for airborne surveys, as discussed in Chapter 6, or used to input any cartographic, photographic or other spatially related data set.

It is this relatively new application of video technology that forms a major part of the data integration methods explored in the development of an operational systems for integration of spatially related data into an easily accessible database.

4.5.1 Framegrabbing

The hardware required for digitising video signals, which essentially is a collection of industry standard chips, is often referred to as a framegrabber.

The framegrabber generally accepts a number of inputs. These vary from a composite monochrome and synchronisation signal, to separate Red, Green, Blue and Synchronisation signals for a full colour framegrab. This form of input generally requires a PAL decoder to split the composite CCIR video signal, output by the camera or video tape recorder, into its components.

The image or frame consists of two interlaced fields; one field being the odd, the other the even lines of the frame. Each set of lines is drawn separately, the speed of the re-draw is controlled by the synchronisation. The CCIR standard operating at 50 Hz redraws a field every 1/50th of a second. Thus 25 images or frames are digitised every second.

This arrangement is similar to the interlacing of a standard television display.

The framegrabber card used in this study, digitised the incoming signal to produce an array of 768 pixels per line with 576 lines. This is well above the resolution of both the camera and video tape recorders used in this study.

4.6 Overlay Production

Using a video camera and framegrabber, or any of the methods detailed above, any form of cartographic data can be digitised. This forms the basis of a spatial data integration system and has a particular application in production of raster overlays containing information on spatial extent of classes, or a vector database of linear features.

For both, the process involves digitisation of the features on the screen. Depending on the method used, the raster digitised representation of the map is generally unacceptable as a final product, due to the radiometric qualities of the digitisation process, and is generally discarded after the features have been extracted.

4.6.1 Spatial Resolution

The general opinion about raster digitised products is that they are not of sufficient accuracy to merit application. This is certainly the case when Ordnance Survey standard digital mapping accuracy is required, but to a large extent the cursory glance afforded by its critics tend to overlook its value in remote sensing applications of limited resolution.

Given that the raster representation of the map is rarely used, and the resolution required for remote sensing studies is limited, raster digitisation can provide a fast, accurate and easy method of input of cartographic data.

As an example let us consider a remote sensing study using the Landsat TM satellite. The study requires that a land use map be produced and overlayed with the several topographic features for reference.

Landsat TM has a spatial resolution of approximately 30m. As the resolution of the final product can not easily be increased above this level, it is pointless to extract information from the map at a significantly higher resolution than this.

The exception to this rule is if future studies are liable to be undertaken with a higher resolution remote sensing input. It might also be considered prudent to extract information at the highest possible accuracy to provide application with future products, or guard against possible inaccuracies when working at a resolution close to that of the study.

Given the availability of video cameras and framegrabbing hardware, and the advantages of video digitisation over scanning systems in terms of areal coverage, a number of the operational factors governing the digitisation process were investigated. These processes could equally well be performed on any source of raster digitised data.

There are several factors controlling the resolution of the digitisation. The area of the map digitised can be altered using a zoom lens or altering the distance of the camera from the map. The area selected depends on the spatial resolution of the study, the total area to be covered and the necessity for discrimination of text and adjacent features.

Two measures were examined to assess the limits of use of video digitisation for input of cartographic data: the legibility of characters, and the separability of adjacent line features.

4.6.1.1 Character Legibility

Table 4.1 and Figure 4.1 show an assessment of the legibility of characters of different sizes on a video digitised image. For the study, text of various sizes was printed at high density on a piece of paper. The font used was a typical map font - Roman. This was then placed on the digitising table under a black and white, security surveillance quality, video camera.

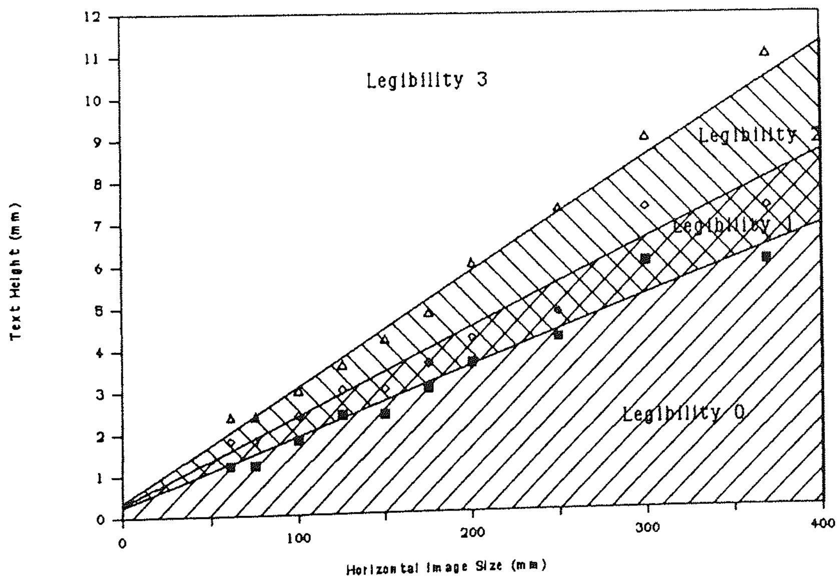
Using the zoom lens the total area digitised was altered. For each setting the size of the area was noted and the raster digitised representation of the text on the monitor was given a legibility rating between 0 and 3. These ratings are subjective and cannot be precisely defined, however these readings can provide rough guidelines as to the size of digitised areas and effective pixel resolution.

Area Size (mm)	Text Size (mm)											
	0.6	1.2	1.8	2.4	3.0	3.6	4.2	4.8	6.0	7.3	9.0	11.0
61	0	1	2	3	3	3	3	3	3	3	3	3
75	0	1	2	3	3	3	3	3	3	3	3	3
100	0	0	1	2	3	3	3	3	3	3	3	3
125	0	0	0	1	2	3	3	3	3	3	3	3
150	0	0	0	1	2	3	3	3	3	3	3	3
175	0	0	0	0	1	2	3	3	3	3	3	3
200	0	0	0	0	0	2	2	3	3	3	3	3
250	0	0	0	0	0	1	2	2	2	3	3	3
300	0	0	0	0	0	0	0	1	1	2	3	3
369	0	0	0	0	0	0	0	1	1	2	3	3

0 The text is illegible, position only determined.
1 The text is just legible, some characters guessed.
2 The text is clearly legible, edges are blurred.
3 The text is clearly legible with defined edges.

Table 4.1. Text Legibility of varying Texts and Digitisation Area.

Limit of Legibility against Size



0 The text is illegible, position only determined.
1 The text is just legible, some characters guessed.
2 The text is clearly legible, edges are blurred.
3 The text is clearly legible with defined edges.

Figure 4.1. Limit of Legibility against Size, see Table 4.1.

4.6.1.2 Line Visibility

The second measure in assessing the limits of use of video digitisation examines the visibility of lines under similar conditions to those used for the legibility of characters. For this study lines of various widths, types and orientation were drawn, placed under the video camera and digitised.

It was found that for the particular camera and lens used the lines drawn were always visible. That is the location and orientation of the line could always be defined, though the type of the line was not always distinguishable. For example line width and style could not easily be determined on the lower resolution digitisation.

There are a number of reasons for this result, all of which depend on the camera and framegrabber used.

The majority of cameras have a built in auto-gradient control, that analyses the light entering the tube, and alter the offset and gain of the circuits to produce an image with reasonable contrast. Digitisation is therefore image dependant with the alteration of contrast making faint lines more visible.

Some cameras and framegrabber modules, like the ones used, have anti-aliasing filters built into their circuits. Anti-aliasing is the process of altering pixel brightness values adjacent to thin lines so that the line does not appear stepped. This has the effect of widening the line, thus making it more visible.

Figure 4.2 demonstrates the effect anti-aliasing has on thin lines. Each pixel on this image is 0.11 mm, the line was 0.5mm wide. The effective width of the digitised line is therefore about 9 pixels - 1mm.

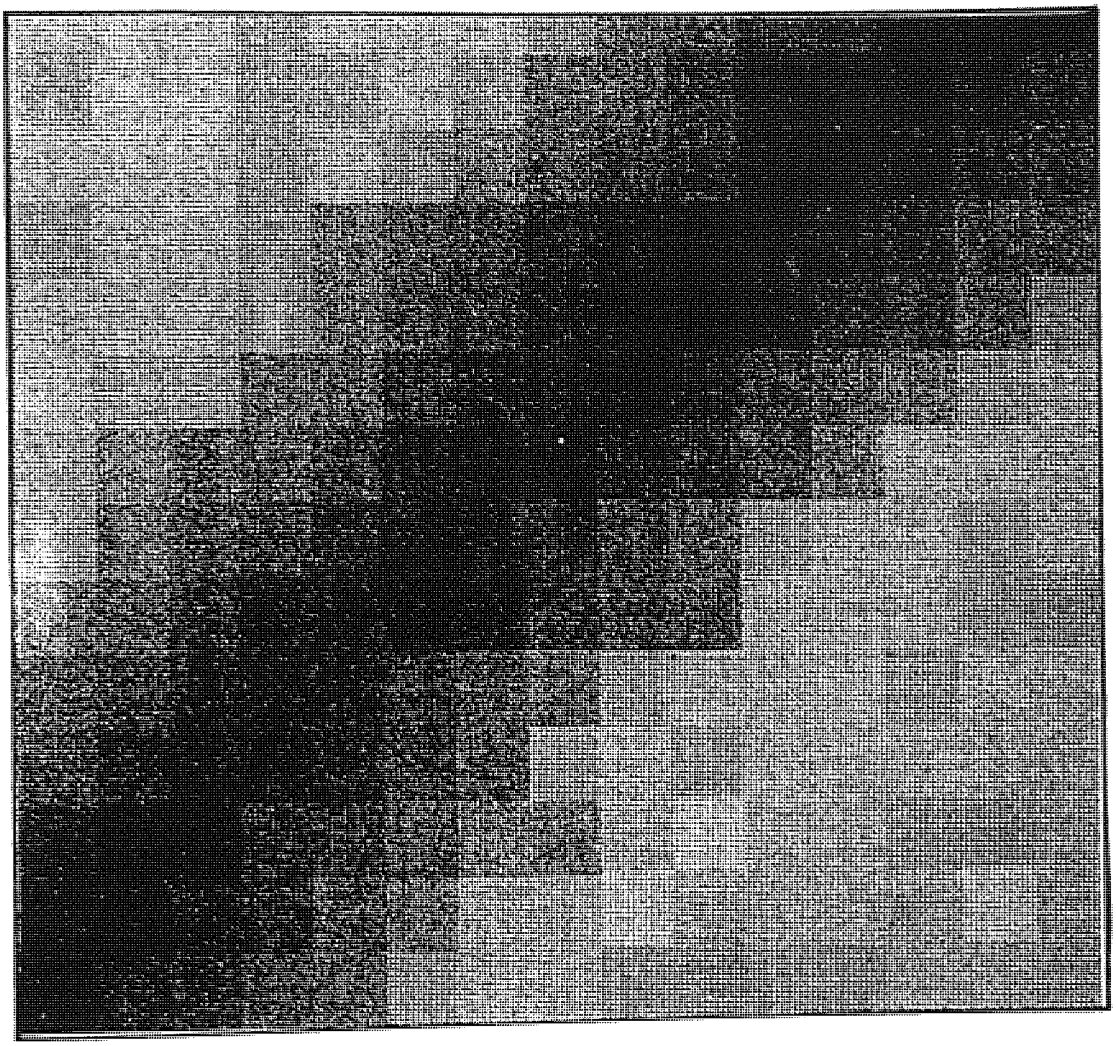


Figure 4.2. The Effect of Anti-aliasing on Line Width.

In addition each pixel's brightness can be sampled over an area larger than the line width. This results in general blurring or widening of the feature, especially where there is a large contrast with the background. In this way narrow lines may appear wider and hence more visible.

For these reasons it was not possible to assess the visibility of lines within the range used. To do this a further test was devised which examined the separability of two lines drawn a certain distance apart at various zoom levels.

The lines drawn are then similar to a number of map features such as roads and buildings. Criteria for selected size of digitised area can then be derived for general position, outline, centre-line or exact edge position.

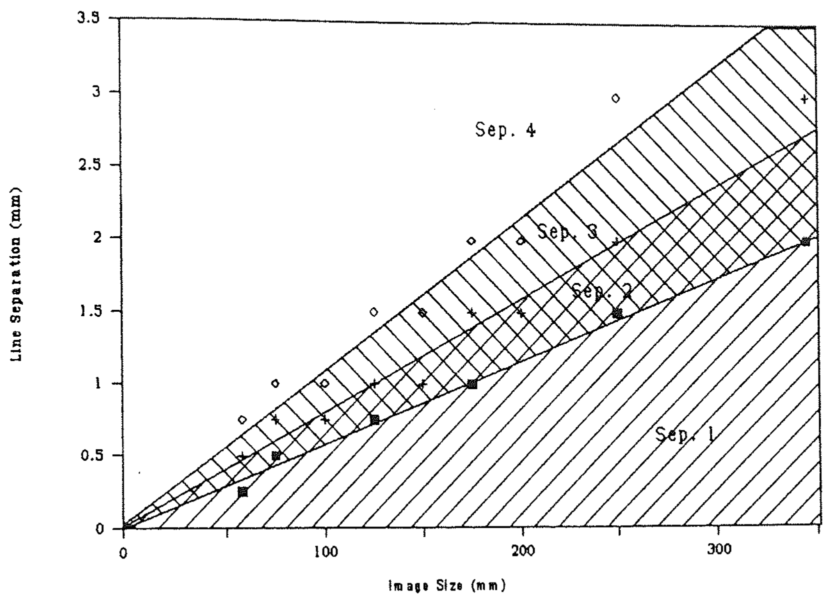
Table 4.2 and Figure 4.3 shows the results of this test. The separability ratings were derived on a similar basis to the text legibility ratings. Again the ratings are mainly subjective, Figure 4.3 can however be used to provide useful guidelines for digitisation area sizes given criteria for line visibility and separability.

	Line Separation (mm)						
Area Size (mm)	0.25	0.50	0.75	1.00	1.50	2.00	3.00
58	1	2	3	3	3	3	3
75	0	1	2	3	3	3	3
100	0	0	2	3	3	3	3
125	0	0	1	2	3	3	3
150	0	0	0	2	3	3	3
175	0	0	0	1	2	3	3
200	0	0	0	0	2	3	3
250	0	0	0	0	1	2	3
345	0	0	0	0	0	1	2

0 The lines are not separable i.e. have merged.
1 The lines are only just separable.
2 The lines are separable although blurred.
3 The lines are clearly separable.

Table 4.2. Line Separability of varying Separations and Digitisation Area.

Limit of Separability against Size



0 The lines are not separable i.e. have merged.
1 The lines are only just separable.
2 The lines are separable although blurred.
3 The lines are clearly separable.

Figure 4.3. Line Separability of varying Line Separations and Digitisation Areas, see Table 4.2.

4.6.1.3 Map Partitioning

The smallest area that can be digitised is approximately 15mm x 15mm. All images are digitised to cover this area with approximately 500 x 500 pixels, giving an effective resolution of 0.03mm, smaller than most lines drawn on maps. With standard zoom lenses the smallest area is about 40mm x 40mm, corresponding to a resolution of 0.08mm.

Obviously the smaller the digitised area, the more areas have to be digitised to cover the whole map. The time taken to perform the various processes involved then becomes a deciding factor in the choice of parameters, and trade-off must occur.

Table 4.3 lists the possibilities for digitising a 10km x 10km (400mm x 400mm) area on a 1:25000 OS Topographic Map.

From this table, which relates to the extraction of features for overlaying on an airborne video survey, it can be seen that at the level of resolution of SPOT HRV data, 20m, the entire 10 x 10 km area can be digitised in one area. This however proves impractical as only a limited amount of data can be distinguished clearly on the image.

Map Area(mm)	Resolution Map(mm)	Ground Area (m)	Resolution Ground(m)	Number of Areas
20x20	0.04	500x500	1.0	400
40x40	0.08	1000x1000	2.0	100
50x50	0.10	1250x1250	2.5	64
80x80	0.16	2000x2000	4.0	25
100x100	0.20	2500x2500	5.0	16
200x200	0.40	5000x5000	10.0	4
400x400	0.80	10000x10000	20.0	1

Table 4.3. Map Area Digitisation Options.

4.6.2 The Process

The various processes involved in the production of both vector and raster databases are summarised in the following sections. The sequence of processes involves raster digitisation, geometric correction, vectorisation of features and production of a raster overlay.

4.6.2.1 Video Digitisation

The first stage in the process is to set up the map under the camera and digitise the selected area. Prior to this the resolution and size of each individual framegrab must be determined. The total map is then partitioned into equal size areas.

Each of these areas is then moved into view of the camera, whilst framegrabbing continuously, and each area adjusted to give a near square image with sufficient contrast to distinguish the required features.

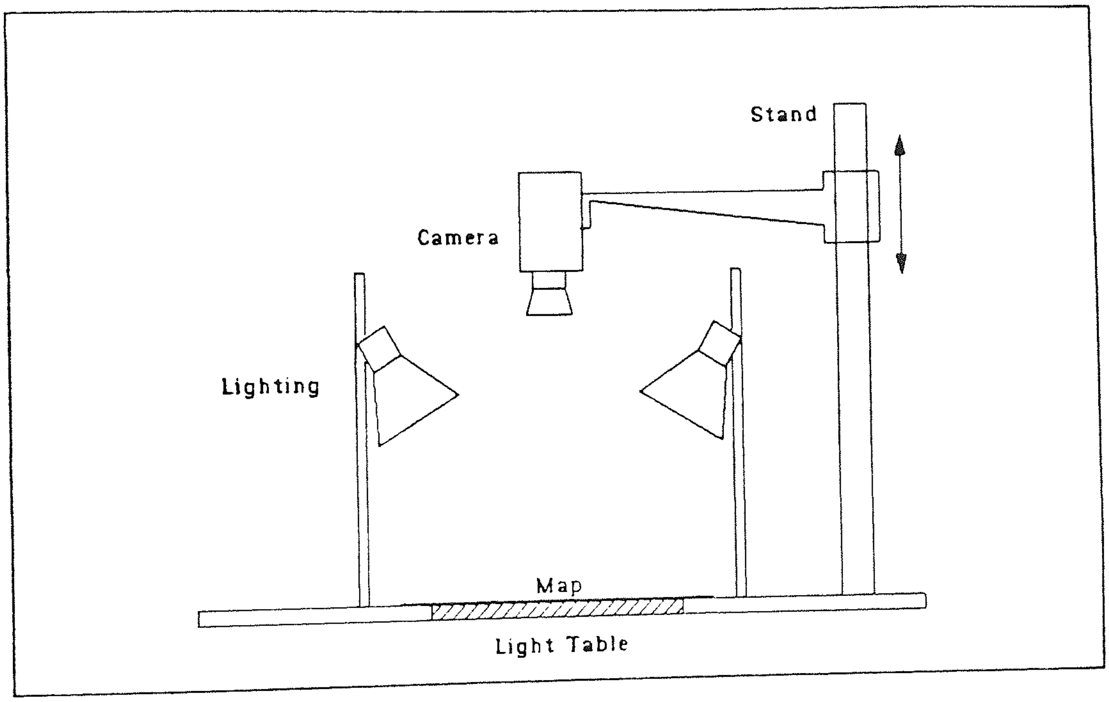


Figure 4.4. Video Digitisation Setup.

The contrast is adjusted by altering the aperture on the lens and ensuring an even lighting situation across the area of interest. Figure 4.4 shows a typical setup for video digitisation of both transparencies using a light table and cartographic products.

For this task once the choice of size and resolution has been made and the total map partitioned, each separate area can be set up, digitised and saved in approximately 2 minutes.

4.6.2.2 Geometric Correction

The necessary next stage is to remove any geometric distortions that may have been introduced at the digitisation stage.

For this process the presence of a grid or a set of recognisable features is required. If not present these should be added to the map prior to digitisation. These points are used to derive a 2-D transformation which will geometrically correct the image or features to a square co-ordinate system.

This is achieved by selecting a number of points for which the map co-ordinates and pixel co-ordinates of the raster digitised map are known. These are then used as input to a linear least squares fit to the data which derive coefficients of x and y, for both u and v the output positions. The process is a standard image processing technique, and in remote sensing applications is often known as warping to ground control points.

The resultant matrix of coefficients describe the total transformation by a combination of scaling, rotation, translation and shear.

At this stage there are two methods for extracting global, corrected co-ordinates. The choice of method depends on the products required.

1. The first method corrects the image prior to the extraction (digitisation) of features. The transformation matrix is inverted, and the output (corrected) image space is defined. For each pixel on the corrected image, the inverted matrix gives the corresponding

position on the input image, and the output pixel brightness value determined using one of a number of re-sampling algorithms on the neighbourhood pixel brightness values.

The choice of the re-sampling algorithm depends on the correction being applied. If the size of the input image is being decreased or kept the same then the fastest algorithm is a nearest neighbour routine. This routine preserves sharpness of edges and lines in the image and so works well for correction of data with limited brightness range. If the image size is being dramatically increased then a bi-linear or bi-cubic re-sampling algorithm should be used. These have a general effect of smoothing the image so edges and lines in the image may become less distinct.

Typical timings for the geometric corrections on the MC 68020 based framestore are between 10 and 50 seconds depending on the re-sampling algorithm, and about 2 min. for definition of the control points.

This method provides two products, feature vertex lists (section 4.6.2.3) and a corrected raster representation of the map, which can be processed as an image.

2. The second method corrects the co-ordinates of the feature as they are being extracted. In this process described in section 4.6.2.3, the pixel co-ordinates are extracted, corrected using the derived transformation matrix, and then converted to global co-ordinates.

The advantage of this method is that the correction is performed during extraction, and hence appears to be instantaneous.

The raster map does not have to be corrected and consequently the only product of the process is feature vertex lists.

The accuracy of the geometric correction is of major importance to the accuracy of the digitised data. The typical accuracy in position is of the order of 1 pixel. Table 4.4 and Figure 4.5 show an example of the accuracy of the video digitisation and geometric correction process.

The example is a 1km square area of a 1:25000 Ordnance Survey Map. Grid Square SS 53 Bottom Left Grid Reference 258000 137000 m.

From Table 4.4, which shows the map co-ordinates, predicted and actual positions of several significant points, it can be seen that the RMS error of the transformation is about 1.6 pixels.

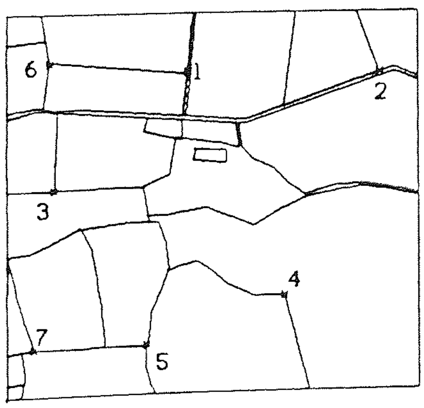


Figure 4.5. Typical OS 1:25000 Field Boundary Data used for check of Accuracy of Digitisation and Geometric Correction, see Table 4.4.

Point No.	Map X	Map Y	Pred. X	Pred. Y	Actual X	Actual Y	Error x Error
1	258445	137840	223	420	222	419	2.00
2	258907	137835	454	417	453	417	1.00
3	258115	137540	56	270	60	271	5.00
4	258680	137250	340	125	339	125	1.00
5	258340	137127	170	64	172	63	5.00
6	258105	137868	53	434	53	433	1.00
7	258065	137120	33	60	34	61	2.00

Table 4.4. Geometric Correction Accuracy.

4.6.2.3 Feature Vectorisation

There are a number of methods of extracting features from cartographic products. At present a large percentage is achieved by manual methods using bit-pad digitisers to follow features on a map pinned to the tablet. This method is well tried and more or less trusted.

Two innovative methods presented here deal with reducing the cost of producing digital map databases, and provide portable and user friendly systems for production and integration of cartographic databases on a project basis. The methods presented are based on manual and semi-automatic methods of digitising features from a raster representation of a map.

The first stage in both processes is to define the position of the window on the global co-ordinate system. This enables features to be digitised in terms of a total or local database co-ordinate system. This includes information for geometric correction of the data.

4.6.2.3.1 Manual Methods

The features are manually vectorised on the image in a similar manner to vectorisation using a bit pad digitiser. That is the cursor is moved to the start point of the feature and set. The feature is then defined as a list of x,y co-ordinates which are the end points of line segments which accurately represent the feature.

Bit pad digitisers output a co-ordinate in a number of ways. These include point mode, every time a button is pressed; stream mode, continuously on a regular time interval; or switch stream mode, continuously whilst a button is pressed. The mode of collection and sampling rates are initially set to the requirements of the user. In stream modes the cursor is 'live' from the moment the start point is set until the end point is set. This is similar to freehand drawing and extreme care must be taken not to deviate from the feature.

The process of vectorisation using an image base has a number of advantages over this method. Instead of a time interval and threshold distance, the user makes his own judgements as to when a point is output. To output a point a button on the device controlling the cursor is pressed. The feature is then defined as a collection of line segments of varying lengths. Each line segment has its start point at the end point of the previous segment and is interactively rubber banded to define the length and end point of each segment. This is similar to the point digitisation mode but has the advantage of leaving a record of the points and line segments defined.

The process can also be paused in the middle of a feature, and any errors caused by erroneous button presses corrected on screen.

The amount of time taken for this stage of the process varies considerably. The time taken depends on a number of factors the most important of which is the number of features to be extracted, and the number of points required to define each feature. This is related to the nature of the study, for example with topographic maps it could be suggested that every feature is extracted into the vector database. Conversely with a geological map, where the areal extent of each geological type is required, only the boundaries need be extracted.

In addition the larger the area covered the more features have to be extracted in each individual digitisation area, and the more difficult it is to extract the features. Conversely the smaller the area the less features have to be extracted and the easier it is to extract the features.

It could however be suggested that, irrespective of resolution, the number of features extracted is constant and therefore the area digitised should be of sufficient resolution to easily extract the features. In this case the other factors controlling time taken in the digitisation process are of more consequence to the total time taken.

Table 4.5 gives timings for extracting a vector database. For this study sections from two 20 x 10 km 1:25000 Topographic maps were digitised in 2 x 2 km areas.

Area Name	No. Pts.	No. Feat.	Time (min.)
bl4642	871	50	55.00
bl4422	534	35	25.00
bl4524	809	43	40.00
bl4724	215	12	15.00
bl4026	424	21	17.28
bl4028	144	5	7.93
bl4228	664	54	38.00
bl4536	216	31	16.85
bl4532	265	26	13.00
bl4530	806	46	35.90
bl4336	154	7	9.25
bl4330	421	49	19.17
bl4332	262	7	7.33
bl4334	60	2	2.88
bl4730	822	76	41.57
bl4732	955	71	39.15
bl4932	660	48	26.23
bl4934	591	48	33.05
bl5132	796	44	34.77
bl5134	509	34	22.90
bl5332	671	53	29.18
bl5334	427	32	25.67
bl5530	598	60	34.55
bl5532	868	97	48.38

Table 4.5. Feature Vectorisation Timings.

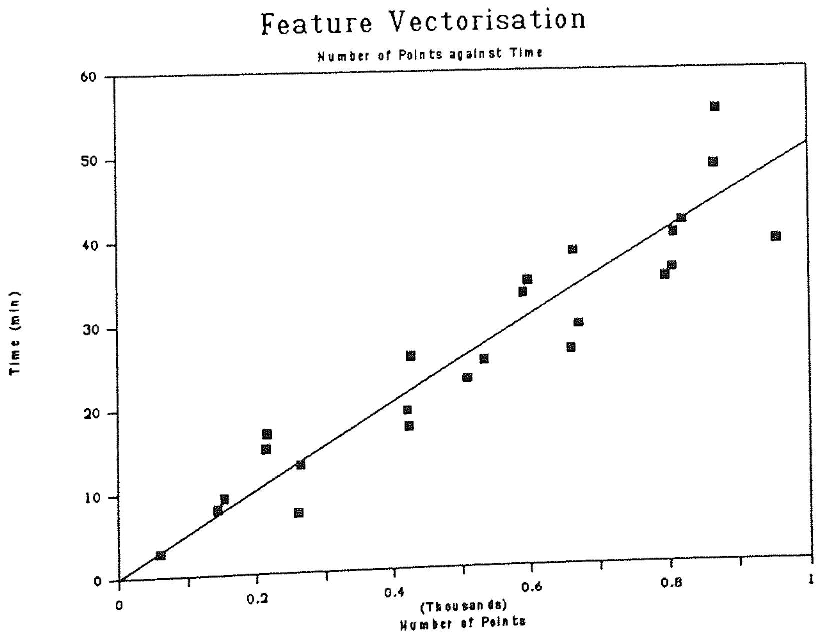


Figure 4.6. Feature Vectorisation Timings, see Table 4.5.

The number of points gives the best measure of the time taken to digitise map features. This is represented in figure 4.6 which shows the time taken to digitise all the features in each 2 x 2 km partition. It can be seen that it takes approximately an hour to create a database of 1000 points.

4.6.2.3.2 Automatic Methods

Full automatic extraction of features into a digital map database requires the algorithm to determine exactly both the feature's attributes and orientation. This is presently beyond the abilities of computer system. There are, however, semi-automatic methods that reduce the overall timings of digital feature extraction.

Perhaps the most fundamental extraction performed on a feature is the determination of attributes, for example line style, colour or contour height. Providing attributes at a manual level significantly reduces the complexity of these algorithms. The user would point to a feature on the raster map, define its attributes, and execute the process for determination of orientation.

Most line following routines are based on a neighbourhood search. From the start position, each pixel's immediate neighbours are searched for a brightness value corresponding to the line. The algorithm then marks the start pixel to prevent a return, writes the position of the pixel found to a list, and moves to that pixel. The process is repeated until all pixels on the line are found or the process requires a decision.

At this stage the line following algorithm must be able to cope with several problems. Some can be solved by preprocessing of the raster map, others by embedding intelligence in the algorithm, or generating queries. Experience has led to the definition of a sequence of preprocessing procedures to aid extraction of features from maps.

1. The first step of the process is to remove any radiometric errors, introduced by variable lighting and lens vignetting, from the raster image. This is achieved by video digitising a monochrome reference card before digitising the map. The image of the reference should contain only the radiometric errors. The mean pixel brightness value for the reference is then determined and the pixel brightnesses of the digitised map transformed by the ratio of the corresponding reference image and the mean reference brightness.

The result of this procedure is an image with pixel brightnesses of each feature roughly constant throughout the image.

2. The second step is to attempt to produce a binary image of the feature to be extracted. With a single feature map, such as a plot of contours and no other features, this is simply a case of determining interactively the threshold brightness. This brightness should be chosen such that the pixel lines are as narrow as possible without breaking the line.

With multiple feature maps, colour can be used to classify the maps, alternatively all features are binarised and classified later in the extraction process. The result of this process is a binary (black and white) image of the required features.

3. The binarised image is then processed using a filter or convolution operation that removes pixels representative of the feature, if their removal does not disconnect pixels of the feature. This process is known as line thinning, and may require several iterations to produce a satisfactory result - thin black lines on a clean white background.

The success of these three operations directly controls the amount of user interaction and intelligence required in the line following algorithm.

If the line thinning operation has removed too many pixels, breaks along the line may occur. The algorithm must be supplied with the intelligence to find the next pixel along the line, across the gap. In this particular case a simple neighbourhood search might jump to a parallel feature of totally different attributes. The routine might generate a query asking the user to identify the feature to connect to. An alternative to this problem, would be to include in the algorithm knowledge of how the orientation of a particular feature type varies. For example, contours are generally smooth and consequently a jump to a feature orientated at an angle greater than 90° is unlikely; road features can have acute angles and a jump to any feature across the gap is therefore feasible.

If the line thinning operation removes too few pixels, the line may be several pixels wide. The algorithm must have the intelligence to know that the line is wide and consequently must only follow the centre pixel.

Junctions in features require special attention. In a semi-automatic system the decision is placed with the user. This accounts for crossings of features, such as road bridges, which again may benefit by use of knowledge of feature types.

Maps with shaded relief, patterned or coloured regions, make the following of features across regions extremely difficult. In these cases a practical alternative is to treat the feature within each region separately.

The problems involved in automatic feature extraction are numerous, each contributes to the manual interaction required by the algorithm. However the time saving is significant in reducing costs, especially with relatively simple maps with a few features. Consequently considerable time and effort will be spent on developing and improving these algorithms in the future.

4.6.2.3.3 Data Storage

Each feature when extracted is given a number of attributes, supplied by the user. These include the colour to draw the feature in, the type of feature, a feature classification code, and the number of line segments that make up the whole feature.

The file of vector data then consists of a set of features each containing an attribute header followed by a list of x,y co-ordinates. This file is searched using the header only and can therefore be described as an indexed sequential file. The adopted vector database format is discussed in Chapter 7.

4.6.2.4 Raster Overlay Production

Using raster digitisation processes there are two main methods of producing a raster database representing thematic information from a boundary map. The choice of method depends on the radiometric fidelity of the raster digitised map, and whether or not the vector boundary data has been digitised.

1. The digitised vector data is recalled and used to define the class boundaries.

2. The raster digitised map processed as described in Section 4.6.2.3.2. This results in a binary representation of the class boundaries.

The boundary of each class is then defined by a line of pixels of a single brightness value. A seed fill process is then interactively performed to mark areas enclosed by a boundary with a unique pixel value. This is repeated until all the areas have been filled.

The advantage of using vector digitised boundaries is that features from several maps can be merged onto a common geographic base.

Once the complete image is created the boundaries can be removed using a convolution filtering technique which replaces the boundary with the median of all neighbourhood pixels not equal to the boundary colour.

The end product is a raster overlay containing only the information relevant to the theme.

The time taken to perform this process is the same irrespective of the size of the digitised area, as it is usually performed on the database as a whole.

4.6.3 Summary

Figure 4.7 is a flowchart of the various stages of the process of producing raster and vector databases using video digitisation.

Given the figures mentioned in the text the total timings for each of the options suggested in Table 4.3 can be added.

Map Area(mm)	Resolution Ground(m)	Number Areas	Time Frame.	Time Correct.	Total (min.)
20x20	1.0	400	800	1200	2000
40x40	2.0	100	200	300	500
50x50	2.5	64	128	192	320
80x80	4.0	25	50	75	125
100x100	5.0	16	32	48	80
200x200	10.0	4	8	12	20
400x400	20.0	1	2	3	5

Table 4.6. Digitisation Area Timings.

In practice it is not feasible to undertake each of these options. Digitisation of the smallest area is too costly in time, the largest area is beyond the limit of text and line visibility, and only a fraction of the required information could be extracted.

Therefore the size of the digitised area is more likely to be a trade-off between the time taken and visibility, which are both to some extent operator dependant.

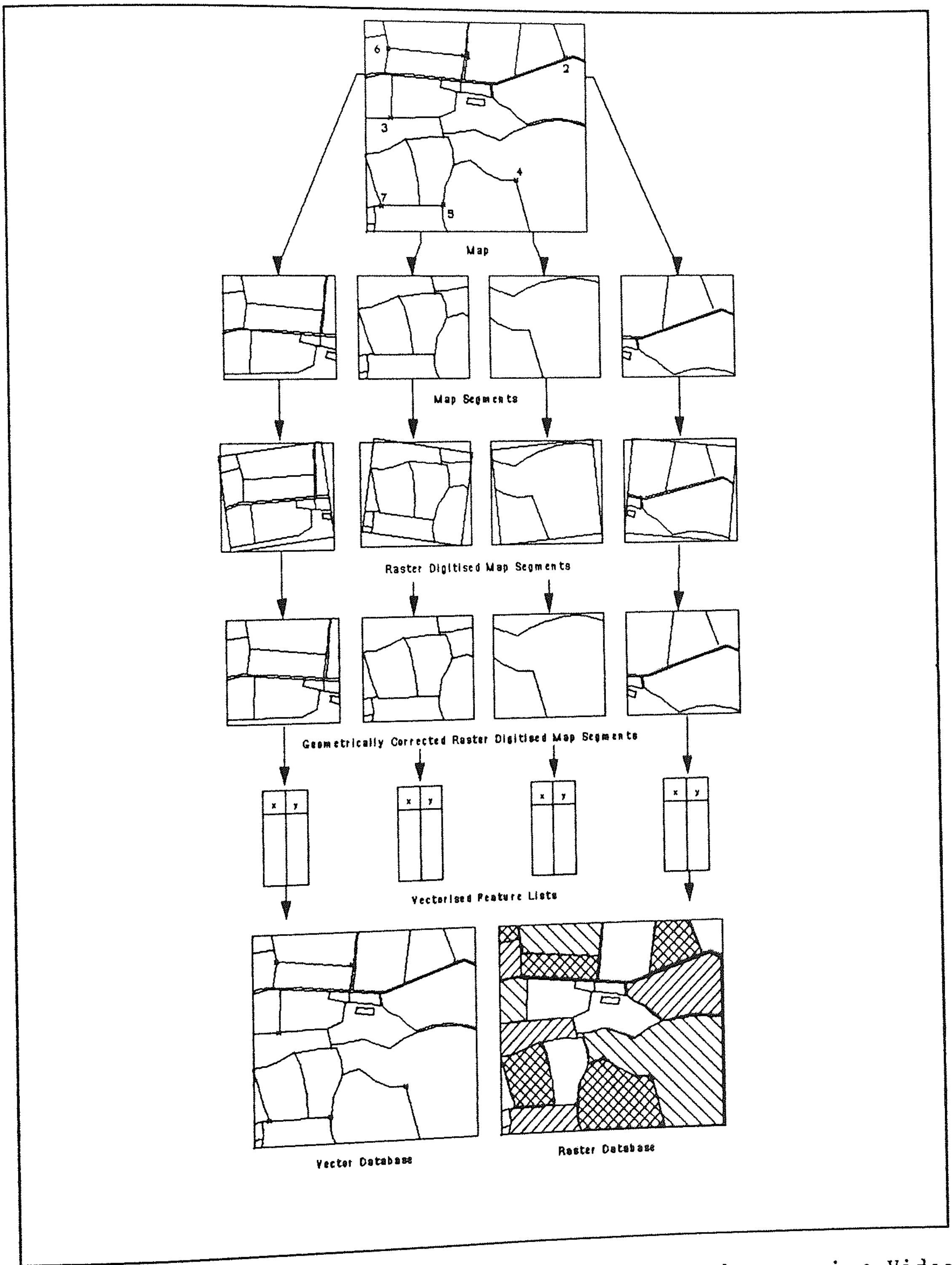


Figure 4.7. The Production of Raster and Vector Databases using Video Digitisation.

5 DIGITAL ELEVATION MODELS

5.1 Introduction

In developing the water information system one form of input, the Digital Elevation Model (DEM), was highlighted more often than any other. A DEM is simply a spatially related array of x-y-z co-ordinates. Its applications range from hydrologic and hydraulic modelling to manipulation of image geometry. Clearly the ability to produce and manipulate this form of data has to be part of any water resources/hydrologically based spatial information system.

The potential for manipulation of DEMs has until recently only been realised by civil engineers in highway design systems. Current spatial analysis systems used for this purpose rely on the use of commercially available DEMs. The Ordnance Survey are producing DEMs of the UK, but in most areas of the world, particularly developing countries, digital topographic data is classified for military use and consequently there is no data available for commercial use. In these situations the user is forced to create the DEM manually, from any available source. This involves digitisation of contours, cross sections or spot height data, which produces a random, or semi-ordered, data set of x-y-z co-ordinates. Given the range of possible inputs and its integration with remotely sensed imagery on a per pixel basis, the elevation data is generally interpolated onto, and stored as, a regular grid. Conventionally, methods of cartographic data integration for DEM production use digitising tablets to input contours, and store the interpolated surface and input data in the computer memory.

This is an example of how image processing technology can be applied in what is essentially a mathematical problem. Framestores can be used as two dimensional spatial arrays to store data, considerably reducing the requirement for host computer memory. Framegrabbing and contour following provides a method of digitising contour and spot height data. This has a particular application where system portability is required as digitising tablets can be bulky. This chapter details the application of methods described in Chapter 4 in the production of Digital Elevation Models. The practicalities of this work is further demonstrated by its application in the production of a multiple theme raster database, including DEM, for the Magat river catchment described in Chapter 8.

5.2 Background

The concept of the Digital Elevation Model was first developed by Prof. C.L. Miller, at the Massachusetts Institute of Technology, between 1955 and 1960, Miller (1957), Miller and Laflamme (1958). Photogrammetry is essentially a method of making spatial measurements, hence measurement of terrain features, particularly elevation, becomes a spatial model. The development of the spatial model concept coincided with the advent of digital computers and automated photogrammetric mapping.

The terms Digital Terrain Model, Digital Ground Model and Digital Elevation Model are synonomous with the spatial model concept developed in this period. Of these, Digital Elevation Model, is used to distinguish between models of elevation and models of other terrain characteristics such as slope, aspect, soils and land use, Doyle (1978), Collins and Moon (1981).

Numerous authors Baussart (1968), Grist (1972) and Doyle (1978), have defined the technology of the spatial model in terms of its components. The Digital Elevation Model is essentially a digital or mathematical representation of the ground surface in terms of x-y-z co-ordinates. Grist (1972) divided the technology into three parts:

a) Data Storage
b) Interpolation
c) Applications

Data storage is analogous to data input to interpolation procedures. Data can be:

a) Ordered	Regular grid of points
b) Semi-ordered	Sections, contours, terrain lines, linked triangles
c) Random	Spot heights, stream intersections

The interpolation procedures required are dependant primarily on the application of the data and secondly on the method of storage. For example many systems use a set of linked triangles as a representation of the terrain, and consequently volume calculations can only be performed if the triangle nodes of two data sets are coincident. If the data is extracted and used in this manner, there may be no necessity for computationally expensive interpolation procedures.

The first documented application of Digital Elevation Models was in the field of highway design, in particular the preliminary design stage of the process. Once the route location has been restricted to a number of possible paths, the DEM can be used as a decision tool to determine which route is the most feasible. This is achieved by the design of vertical curves and final orientation to minimise earthworks. This requires calculating necessary cut and fill volumes from the terrain and proposed highway design. The DEM is therefore a tool for location, design and quantity analysis of highways, railroads, canals, levees, dams, dredging and general civil engineering works. This particular application forms the basis of design systems such as MOSS and ECLIPSE.

Further documented applications include contouring, profiling, intervisibility studies, perspective viewing, navigation control systems (ICBMs), terrain simulation, slope and aspect, shaded relief and use in hydrologic and hydraulic computer simulation models. Some of these applications are detailed in Section 5.4.

5.3 Interpolation

The process of interpolation provides a method of modelling a surface, mathematical or otherwise, in such a way that the surface fits the original data exactly, and provides a logical interpolated surface where data does not exist.

In this application the interpolation procedures use a random array of points on a surface as input and produce a regular, rectangular, grid of points. In the production of a Digital Elevation Model this manifests as using significant points of topographic interest, and generating a surface that gives the elevation of any point within a sepecified area.

Two main sorts of interpolation have been used, Weighted Least Squares and Multiquadric Interpolation. These differ in their nature: least squares is a point, multiquadric a surface, interpolation. Both produce comparable results in terms of accuracy, but differ considerably in time of computation and storage required.

Other interpolation procedures considered, but not implemented, include the triangulation procedures adopted by surface modelling systems such as MOSS, interpolation using finite elements, Ebner (1984), applications of kriging to surface interpolation, Ellehoj and Lodwick (1983), Delfiner and Delhomme (1975), and interpolation using weighted averages, Yoeli (1975).

5.3.1 Data Input

The primary source of elevation data available today is on topographic maps of varying scales, and invariably in the form of contours with a few scattered spot heights.

Using video digitisation the contour map can be input, and the contours extracted. The extraction process generates a stream of x-y-z co-ordinates where each co-ordinate corresponds to the end point of a segment of the line or contour. The density of the points extracted from the contour is dependant on the nature of the ground surface.

As can be seen from Figure 5.1 the nature of contour digitisation produces a distinctive form of input to interpolation procedures. This form of input is not entirely random as it has order and certain features of this input should be noted. The input has a very high point density along the contours. Conversely the input has a very low or zero input between the contours.

Hardy (1971) suggested that the ideal form of input to an interpolation are significant points of topography such as peaks, depressions, saddles, significant changes in slope and junctions of drainage patterns. This is the form of input derived from original survey data and is generally the minimum necessary to define a terrain. Unfortunately this input is rarely available, and it is therefore necessary to use the contour map as the primary source of data.

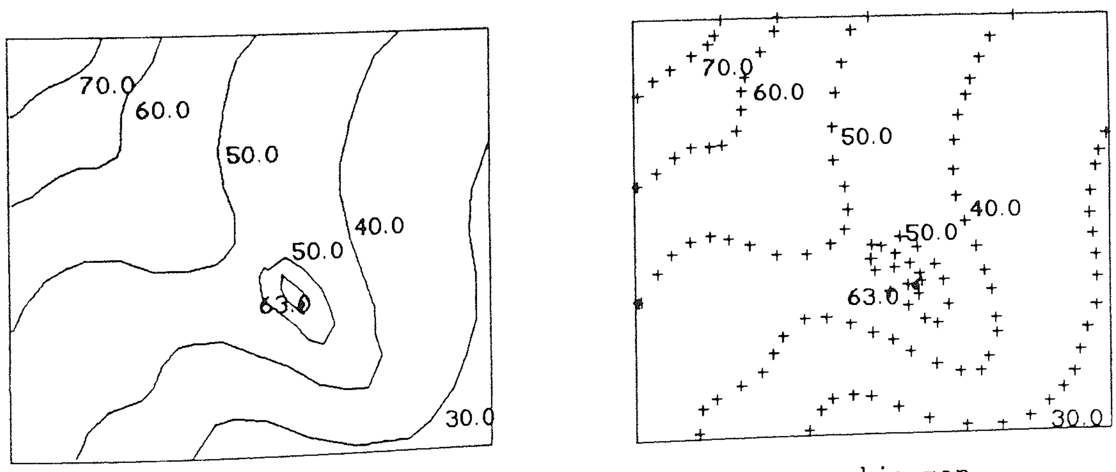


Figure 5.1. (a) Representation of contours on topographic map.

(b) Elevation data extracted from contours.

If all points are given equal weighting, using digitised contours results in a surface markedly stepped in appearance. This is caused by the local search of the nearest n number of points finding only 1 elevation for the area - that of the nearest contour.

The first solution to this problem is to analyse the topography depicted by the contours and derive input data only at significant points. To do this the user must be able to visualise the surface topography, and therefore requires a certain amount of interpretative skill. This solution does not lend itself to fast input of data, although it may provide the quickest and most accurate input when considering the additional computation required to achieve the desired product.

The second solution is to reduce the number of points along the contour so that the density is roughly constant along and between the contours. This can be visualised as a totally random input where the orientation of the contours could not be derived, i.e. the data has no trends.

A subset of this second solution is to reduce the size of the interpolated surface so that the point density is very high throughout. The full surface can then be produced by bi-linear interpolation from what is essentially a coarse grid surface to a fine grid. This process causes a lot of the fine variation to disappear, which may or may not be advantageous.

The final solution is to derive search algorithms that search for a minimum number of points in all directions. This will then find these points in directions perpendicular to the contour line i.e. the adjacent contours, as well as along the contour line.

5.3.2 Hardware Usage

In conventional interpolation procedures, the input data is held in an array which is not spatially related. That is, the input data is three 1-dimensional arrays holding the x,y and z co-ordinates of each point. This array is sorted (and re-sorted) so that the nearest points to any point can easily be found. Alternatively, some routines hold an array the size of the interpolated surface, with the input data marked on sorting, to obtain their spatial relationship. This requires large amounts of storage as the input and output data must be preserved throughout. The output for these procedures is generally a large array, dimensioned in RAM, equivalent to the size of the resultant surface.

On image processing systems large amounts of spatially related storage is available. This memory is in the form of dual ported video memory - a framestore. In the system used, see Section 3.4.3, the input to the interpolation is held in an 8 or 16 bit component of video memory. This component gives the input values, x and y from the screen position and z from the brightness value. The spatial relationship of the points is also easily derived.

The output from the interpolation is then represented by a second component of the video memory. The interpolated values are directly output to the video memory giving an immediate visualisation, during operation, of the resultant interpolation.

A number of interpolation routines were developed to use the hardware features of the image processing system. All were developed from basic principles in compiled Microsoft 'C', and tested on a Jarogate Sprite Intel 80386 based AT compatible operating at 16.67 MHz with an Intel 80287 Math Co-processor.

5.3.3 Least Squares

Least squares interpolation procedures work by fitting a mathematical surface through a specified number of points such that the squares of the deviation from the surface is a minimum.

The order of the surface used can vary from a simple plane surface to higher order polynomial surfaces.

5.3.3.1 The Algorithm

The particular algorithm used fits a surface to a local input for each point on the surface (pixel). The process is a point rather than a surface interpolation, where the solution of the least squares fit produces a value for just one point. Least squares have long been a standard method of interpolation, and published accounts of its application are numerous. For example Junkins and Jancaitis (1972) described its application in production of smooth contours in 2-dimensional space, Kraus (1972) and Arthur (1973) used least squares to account for planimetric distortion in photogrammetric mapping. The least squares surface interpolations implemented and described in this text are detailed more fully by Schut (1974), Hardy (1977) and Allam (1978).

An enhancement of the least squares interpolation produces local surfaces that combine to give a total surface. Much of this work has been published by Junkins and Jancaitis (1973,1974). These local surfaces are sufficiently small to model with lower order surfaces, the total surface at each edge is controlled such that the partial surfaces join seamlessly. The same authors also introduced a general purpose surface averaging concept, whereby the partial surface is the weighted average of four larger preliminary surfaces, overlapping the final surface. This procedure provides the ability to vary the smoothness of the final surface by increasing the size of the preliminary surfaces. The extra processing required and increased complexity of the algorithm would not seem to outweigh the benefits, consequently this particular form of least squares interpolation was not adapted.

Once the method has been defined the basis of the algorithm is standard.

The general equation of the surface is given by :

$$h = a_0 + a_1 x + a_2 y + a_3 x y + a_4 x^2 + a_5 y^2 \ldots\ldots$$

For a plane surface this reduces to :

$$h = a_0 + a_1 x + a_2 y$$

Bi-linear Polynomial (hyper - surface) :

$$h = a_0 + a_1 x + a_2 y + a_3 xy$$

Second Order Polynomial :

$$h = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

For each of the n points found in the locality this equation is applied, giving in matrix notation :

$$\begin{bmatrix} 1 & x_0 & y_0 & x_0 y_0 & x_0^2 & y_0^2 \\ 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & x_{n-1} & y_{n-1} & x_{n-1} y_{n-1} & x_{n-1}^2 & y_{n-1}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ . \\ . \\ z_{n-1} \end{bmatrix}$$

which corresponds to :

$$AV = Z$$

In addition each point is weighted to scale its contribution to the interpolated surface, in matrix notation :

$$W = \begin{bmatrix} w_0 & . & . & . & . \\ . & w_1 & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & w_{n-1} \end{bmatrix}$$

where each element is given by the reciprocal of the square of its distance from the point to be interpolated :

$$w_i = \frac{1}{\left(\left(x - x_i\right)^2 + \left(y - y_i\right)^2\right)}$$

Using standard least squares procedure :

$$AV = Z$$

becomes :

$$A^T W A(V) = A^T W Z$$

and the solution is :

$$V = \left(A^T W A\right)^{-1} A^T W Z$$

Depending on the order of surface this gives the solution of m equations with m unknowns, where m is related to the order of the surface. For example a plane surface gives the solution of 3 equations in 3 unknowns. The unknowns are then substituted into the original surface equations to yield the elevation for that point.

5.3.3.2 Sorting

For each point on the proposed surface the above process is performed. The first stage in this process is to determine the points that are included in the surface fit.

To do this an initial search distance is specified and the algorithm then searches an area approximating a circle of radius equal to the initial search distance. If the number of points found is greater than a specified minimum, and less than the limit placed on the input array, then the surface is fit and the elevation output. If the number of points is outside these limits the search distance is altered and the area re-searched.

Experience with interpolation of a number of test surfaces suggests 8 initial points provides a relatively fast and sufficiently smooth interpolation. The initial search distance is then chosen on examination of the spatial relationship and density of the input data, such that the test for a minimum number of points is met on the first search.

To account for the nature of the input, digitised contours, special search algorithms were derived. These routines divided the input space into quadrants, or octants, centred at the point to be interpolated. Each of these quadrants are then expanded so that a quarter of the minimum number of points is found in each. This slows the search process considerably but produces a more logical interpolated surface.

5.3.4 Multiquadric

Multiquadric interpolation routines define the interpolated surface as a summation of a series of local surfaces. Each local surface is a mathematical function such as a hyperbola or parabola. Each input data point is used to derive an array of coefficients which determine how much of each of the local surfaces contributes to the total surface. The interpolation is performed by deriving these coefficients, and substituting into the formula for the local surfaces.

The main difference between this and the least squares interpolation is that every point is fitted exactly and the resultant interpolation is a surface not a point.

Published work on multiquadrics by Hardy (1971,1977) and Schut (1974) suggested the optimum function for the local surface was a cone or sharp nosed hyperbola. However with both soft nosed hyperbole and the parabola severe oscillation occurs, resulting in a loss of accuracy (see table 5.4), and these functions were therefore not used.

5.3.4.1 The Algorithm

The term multiquadric refers to a series of quadric functions. The following could equally well be applied to multi-cubic and multi-quartic series.

The general form for a multiquadric series is given by :

$$\sum_{n}^{i=1} Q_i\left[q(x_i, y_i, x, y)\right] = z$$

For a hyperbola the surface is defined by :

$$\sum_{n}^{i=1} Q_i\left[(x_i - x)^2 + (y_i - y)^2 + C\right]^{\frac{1}{2}} = z$$

Setting C = 0 gives a series of Cones.

For a parabola the surface is defined by :

$$\sum_{n}^{i=1} Q_i\left[(x_i - x)^2 + (y_i - y)^2 + C\right] = z$$

Thus for each local surface a number of points are found, for each point a quadric function, and the resultant surface represented by a summation of these functions.

In matrix notation :

$$AQ = Z$$

where A is the symmetrical matrix of coefficients of x and y :

$$A = \begin{bmatrix} a_{0,0} & a_{1,0} & \cdot & a_{i,0} & \cdot & a_{n-1,0} \\ a_{0,1} & a_{1,1} & \cdot & a_{i,1} & \cdot & a_{n-1,1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{0,j} & a_{1,j} & \cdot & a_{i,j} & \cdot & a_{n-1,j} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{0,n-1} & a_{1,n-1} & \cdot & a_{i,n-1} & \cdot & a_{n-1,n-1} \end{bmatrix}$$

each element of which is given by :

$$a_{i,j} = \left[(x_j - x_i)^2 + (y_j - y_i) + C \right]^{\frac{1}{2}}$$

Q and Z are the matrices of unknowns and input z values :

$$Q = \begin{bmatrix} q_0 \\ q_1 \\ \cdot \\ q_i \\ \cdot \\ q_{n-1} \end{bmatrix} \qquad Z = \begin{bmatrix} z_0 \\ z_1 \\ \cdot \\ z_i \\ \cdot \\ z_{n-1} \end{bmatrix}$$

The solution is then given by :

$$Q = A^{-1} Z$$

The height for every point on the local surface is then calculated by substituting these coefficients and the x and y position of the point to be interpolated into the quadric equation.

On examination of the coefficient matrix A, two important factors should be noted. Firstly, the matrix is symmetrical, and consequently only half the matrix needs to be derived. Secondly, if the constant C is set to 0, as in the case of the cone, then the leading diagonal elements of the matrix are also 0.

This can cause problems with the equation solver. The simpler forms of equation solver search for the pivot on the leading diagonal and therefore fail in this situation. It is therefore necessary to use an equation solver that re-orders the matrix to find the best pivot value, normally the maximum value on each row. The method applied in the algorithm is Crouts LU Decomposition method with partial pivoting. An alternative method that could be used is Singular Value Decomposition or SVD. These equation solvers are well documented and source provided in numerous texts, Press (1988).

5.3.4.2 Sorting

It is clear from the description of the algorithm that every point contributes a row and a column to the coefficient matrix. Thus with only a relatively small number of points, a very large matrix is generated and has to be inverted to provide the surface.

To interpolate the entire surface, the ideal situation would use the entire data set, produce and invert one coefficient matrix, and produce the surface in a single operation. On a desktop machine the memory limits corresponds to a limited number of points, and therefore the solution must be segmented.

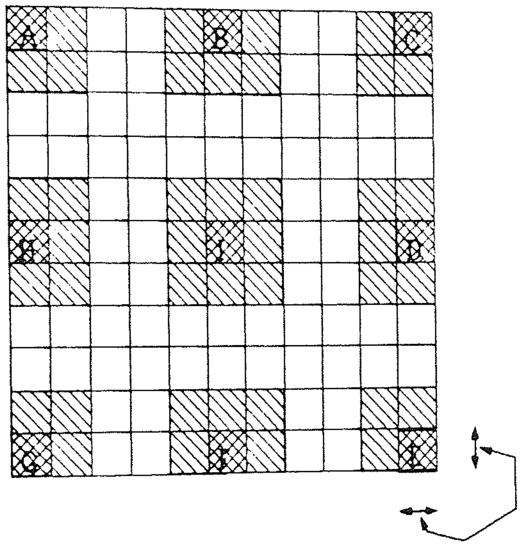


Figure 5.2. Multiquadric Surface Segmentation, showing Total and Partial Surface, and 9 possible cases of input initial search area.

Initially the interpolated surface was divided, and each sub-surface was obtained using the points falling within the area. This provides no means of ensuring the sub-surfaces match at the edges. To account for this a sort algorithm was derived that searched a larger area around the sub-surface for the input data, but only produced a surface inside the area. Figure 5.2 shows the interpolated surface and subsurfaces, with the input search area for all possible cases at the edges, corners and internal areas of the total surface. This is analogous to the surface averaging concept developed by Junkins and Jancaitis (1973,1974), in ensuring smooth boundaries between sub-surfaces by deriving each surface from a larger area.

If the number of points found exceeds the system limit or is less than the minimum number of points required to define the surface, then the input area is contracted or expanded until the limits are met.

This generally produced good results, only failing where there was little or no input (as with any interpolation!).

5.3.5 Comparison

The two implemented algorithms can be compared in terms of storage requirements, speed of execution and accuracy.

5.3.5.1 Storage Requirements

Given that the process is performed on an IBM PC-AT with maximum 640Kbyte of RAM, the temporary storage required by the process is an important factor in choice of a number of parameters.

Using the framestore to store the input data in a spatial representation and display the output from the interpolation the only memory requirements are temporary, and consist of the local surface and coefficient information.

For each point the least squares routine finds the nearest n number of points. These are then held in three 1-dimensional arrays of size n, X,Y & Z. The array of coefficients of x and y is derived and stored in a m by n array, A, where m is related to the order of the surface. For example a plane surface fit to 8 points - a 3 by 8 array.

The weights are then derived and, although represented by a 2-dimensional array above, are stored in a 1-dimensional array of size n, W.

Several temporary arrays are used in the matrix multiplication. The resultant matrix is a m by m array, which is then inverted to give the solution.

The total memory requirements for a least squares mth order surface fit with n points is give by:

X,Y,Z	3 x n	Float	
A	m x n	Float	
W	1 x n	Float	
WS	1 x n	Float	
WA	m x n	Float	
ATWS	m	Float	R.H.Side for equation solver
ATWA	m x m	Float	L.H.Side for equation solver
Output	1	Integer	
Total	((5 + 2m)n + (m+1)m) x 8 bytes + 1 x 2 bytes		

The memory requirements for a plane surface fit to 8 and 100 points are 802 and 8,898 bytes respectively. It is therefore clear that, if memory requirements were the only concern, a large number of points could be included in the surface fit for each point, and a higher order surface could be used.

For a multiquadric surface the routines finds all points within the partial surface and overlap area and stores the data in three 1-dimensional arrays: any number of points can be found within the search area. The matrix of x-y coefficients is then derived. This matrix is held in a n x n array of floating point elements. On inversion this matrix is destroyed and the result stored in the solution matrix Q of size n.

The total memory requirements for multiquadric interpolation with n points to a p x q partial surface is give by:

X,Y,Z	3 x n	Float
A	n x n	Float
Q	1 x n	Float
Output	p x q	Integer
Total	((4 + n)n) x 8 bytes + (p x q) x 2 bytes	

Typically for a 20 x 20 partial surface using eight points - 896, and a hundred points - 11200 storage locations.

The memory requirements for a partial surface interpolation of size 20 x 20 from 8 and 100 points are 1568 and 84,000 bytes respectively. The requirements increase rapidly with the number of points. Clearly the number of points used quickly becomes a concern with the limited memory on PCs.

Of the two processes multiquadric interpolation requires more storage, but interpolates a surface rather than a point. This has a significant effect on the speed of execution. Provided the partial surface is relatively small, and the number of input data points low the memory requirements are not restrictive.

5.3.5.2 Timings

Perhaps of more concern to desk top computer users is the speed of operation of a process. Micro-computer users expect an immediate response from a dedicated system, and therefore speed becomes an important factor.

There are two processes which control the speed of execution of interpolation procedures. The first is the time taken to search the input space for the requisite number of points. This depends to a large extent on the initial search area which should be chosen to find the minimum number of points (and no more) in the first iteration. The second process is that of producing the x-y coefficient arrays and its subsequent inversion.

With least squares interpolation the speed of inversion of the matrix is constant and depends on the order of surface. Using lower order surfaces increases the speed of execution. The time taken to reduce the x,y,z co-ordinates to the matrix to be inverted is dependant on the number of points. It is essential that both these processes are optimised, as the process is repeated for every pixel on the resultant surface.

With the multiquadric interpolation the time taken to invert the matrix derived is approximately proportional to the square of the number of points. Therefore the segmentation of the interpolated surface, directly affects the speed of operation. The larger the partial surface used the more points the search will find, the more likely that the algorithm will have to decrease the search size and re-search, and the longer the inversion process will take. If the partial surface is small, less points will be found, but it is more likely that the algorithm will have to increase the size and re-search. More importantly, the smaller the partial surface the more searches and inversions will have to be carried out to cover the same area. Clearly the choice of the partial surface size is a prime consideration.

Tables 5.1,5.2 and Figure 5.3 shows the relationship point density has with time, for both interpolation procedures. These tables corresponds to a varying interpolated surface with a constant number of points rescaled to fit the surface. Clearly the higher the point density the quicker the interpolation. This can be explained by there being less data to interpolate. In addition the input data is found quicker without redefinition of the search area and re-search of the input space.

The optimum initial search radius for the least squares was found to be 1 for these relatively small surfaces and high point densities. The optimum partial surface for the multiquadric interpolation was found to be approximately 1/10th of the total size in 1 dimension. That is for a 100x100 surface the partial surface or interpolation was carried out on a 10x10 area. To cover the total area this was then repeated 100 times.

Size	Pts.	Ps/sq.px	Rad	Time(s)	RMS Err	Max Err
40	504	0.315	1	71	1.18	5.00
50	528	0.211	1	128	1.04	4.00
60	526	0.146	1	222	0.94	4.00
80	533	0.083	1	605	0.81	3.00
100	533	0.053	1	1456	0.74	2.00

Table 5.1. Point Density against Interpolation Time, Least Squares. Figure 5.3.

Size	Pts.	Ps/sq.px	PI	Time(s)	RMS Err	Max Err.
40	504	0.315	4	496	0.67	1.00
60	526	0.146	6	607	0.63	1.00
80	533	0.083	8	698	0.65	1.00
100	533	0.053	10	783	0.66	1.00
200	525	0.013	20	1459	0.66	1.00
500	518	0.002	50	6278	0.56	1.00

Table 5.2. Point Density against Interpolation Time, Multi-quadric. Figure 5.3.

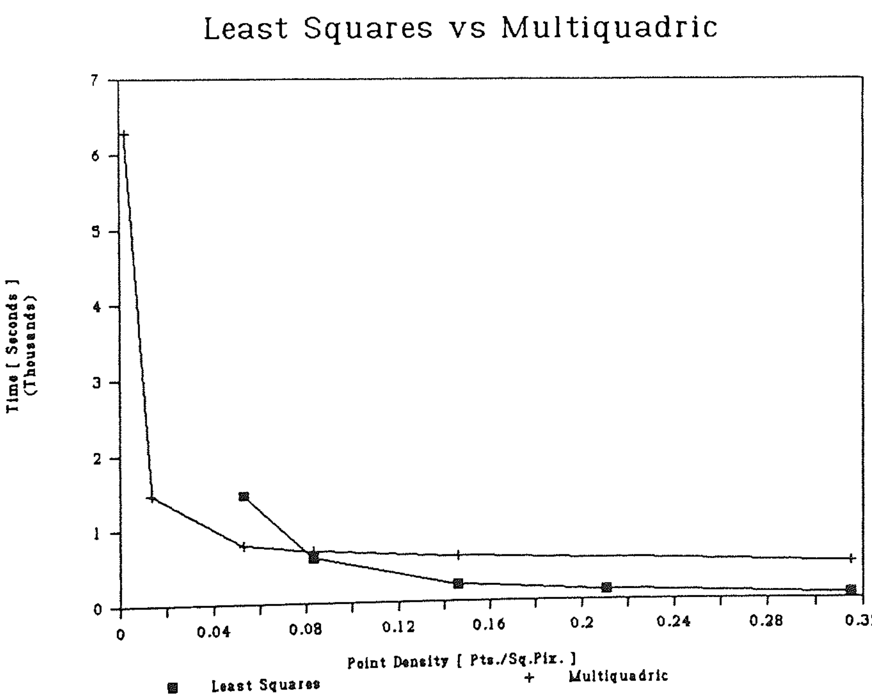


Figure 5.3. Point Density against Interpolation Time. Tables 5.1,5.2.

Table 5.3 shows the results of a comparison of three models of least squares that of the plane, bi-linear and 2nd degree polynomial surfaces. As expected the extra computation involved in the higher order surfaces increases the interpolation time by about 50% per order of surface.

Model	Time (s)	Input RMS	Input Max.	Contour RMS	Contour Max.
Plane	192	0.99	3.00	1.08	4.00
Hyper	241	0.97	3.00	1.07	4.00
2 Poly.	379	0.80	3.00	1.00	4.00

Table 5.3. Least Squares Surface Models.

On examination of the theory of multiquadrics it can be seen that altering the constant C, effects the shape of the surface used in the series. For example it has been seen that with C=0.0 the surface is a circular cone, as C increases the shape of surface moves towards sharp nosed hyperbole, Figure 5.4. As can be seen from Table 5.4 this does not significantly alter the interpolation time, as expected, however the accuracy decreases rapidly as C increases.

C	Time (s)	Input RMS	Input Max.	Contour RMS	Contour Max.
0	410	0.58	1.00	1.38	5.00
10	412	0.61	1.00	1.05	5.00
50	412	1.09	12.00	1.53	15.00
100	414	3.73	42.00	3.94	42.00

Table 5.4. Multiquadric Surface Models.

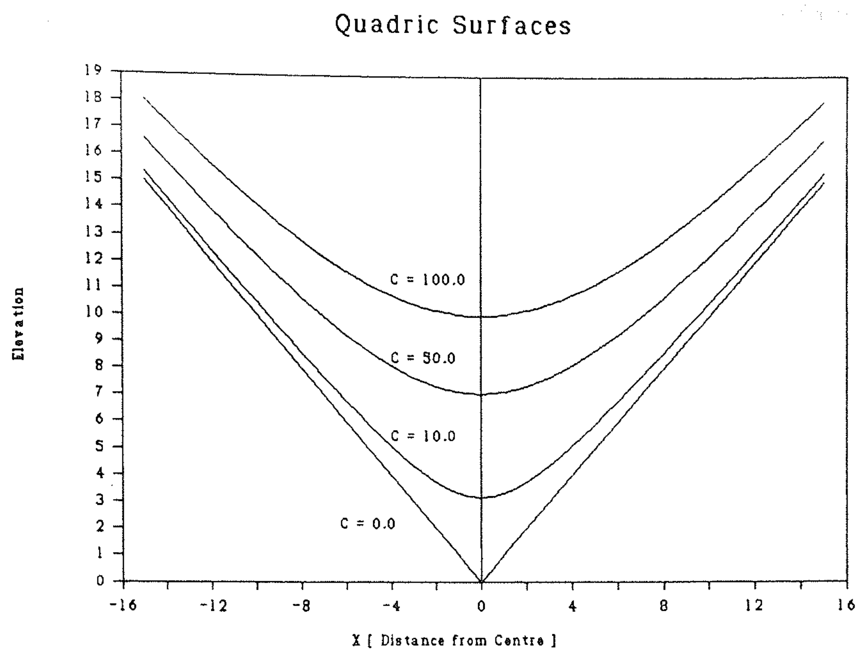


Figure 5.4. Quadric Surfaces from Cones to Hyperbole.

5.3.5.3 Accuracy

The accuracy of the interpolation was checked by comparing the input and output at the position of the input data points and at points between the input data along the contours. The accuracy measured, in terms of root mean square and maximum error is presented in tables 5.1 through 5.4.

On examination of Tables 5.1 and 5.2 it can be seen that the lower the point density the higher the accuracy of the interpolation. This may partly be explained by the relaxation of control of the individual surface fits as the points move further away from the point to be interpolated, thus giving the closest points a higher relative weight.

Table 5.3 demonstrates that the higher the order of surface used in least squares interpolation the more accurate the surface fit.

A combination of these results suggest that the highest accuracy for least squares is given when using the highest order surface, lowest point density and lowest initial search area. This also happens to be one of the slowest options, and clearly trade-off between accuracy and time has to occur.

It is worth noting at this point that to some extent the interpolation is data dependant. That is, a smooth rolling topography, as with this study, is best modelled using higher order surfaces, Conversely rough fractured topography is best modelled using lower order surfaces. This is pictorially represented in Figure 5.5.

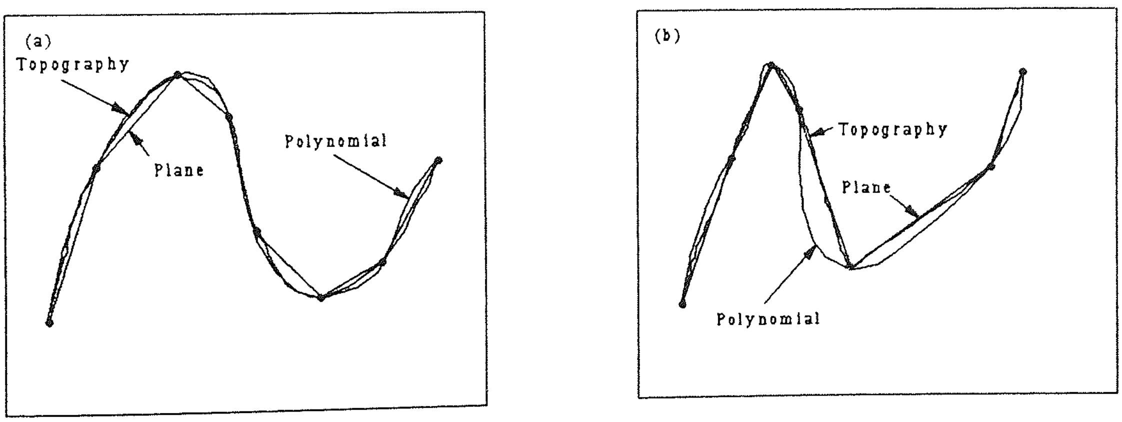


Figure 5.5. Surface Approximation using Mathematical Functions. (a) Smooth surface (b) Rough Surface.

It is demonstrated in Table 5.4 that the cone is close to the optimum surface type for accuracy in multiquadric interpolation. Although with the contoured input a sharp nosed hyperbola with C=10.0 produced more accurate results.

Therefore the overall accuracy of the interpolations can be seen as comparable. There is however one further point of note. The accuracy of the least squares interpolation is data dependant. The accuracy for the test case refers to a smooth rolling topography, the same surface could not as accurately depict a rough terrain. With the multiquadric interpolation the accuracy is not as data dependant and the accuracy does not alter significantly in different terrains.

5.4 Applications

There are a number of applications of Digital Elevation Models which can make use of standard image processing techniques and are therefore suited to DEMs produced by the methods detailed in the previous sections.

5.4.1 Height Representation

Conventionally contour maps have been used to represent 3-dimensional height variation on 2-dimensional topographic maps. The contour describes the shape of each feature, the closeness of contours the steepness of the topography. Contour maps can also be easily produced using x-y plotters, and consequently will probably be used for years to come. There are other conventional methods of representing height, such as sections and shaded relief, and several computer derived representations, such as height surface maps and simulated perspective views.

Cross sections and longitudinal sections represent the variation of height along a line of interest, and are used in representation of linear features such as roads and rivers. Both these representations are easily derived using image processing techniques.

The Digital Elevation Model is represented on the display as a height surface map. The brighter the image pixel or ground element, the greater its height. This in itself provides an easily comprehended representation of the terrain.

The production of a contoured representation of the terrain from a height surface image is achieved using what is known as a density slice. This operation colours ranges of pixel brightness values (heights) to a specified colour. Multiple ranges are defined in different colours producing a height interval map. The boundaries of each interval are the contours, which can be extracted using simple boundary detection algorithms such as high pass filters.

Sections are easily produced by reading the brightness values from the framestore along a raster converted vector. The offset along the section is given by the euclidean distance, the height determined from the pixel brightness.

Shaded relief maps and perspective views are described in sections 5.4.5 and 5.4.6.

5.4.2 Volume Calculations

The calculation of volumes using Digital Elevation Models represented in this manner, involves the calculation of the difference in image brightness (height) between two height surfaces.

This is achieved by subtracting one DEM image from another. The resultant image is a display of positive and negative height differences, the difference in height being given by the brightness value, the volume being calculated by locally summing the differences and multiplying by the pixel grid size.

This has obvious application for all volume quantity analysis work, such as highway design, dam construction and harbour dredging.

5.4.3 Slope and Aspect

Slope and aspect are two measures with numerous application in integrated information systems. Typical examples are analysis of erosion risk and possible siting of solar panels for housing projects.

There are a number of algorithms available to derive slope and aspect. Aspect can be defined as the direction of the greatest slope. With DEMs stored as linked triangles the slope and aspect refer to the area enclosed within each triangle and are easily determined from the node heights. With raster based DEMs the slope and aspect are derived on a per pixel basis, from examination of neighbourhood values.

The simplest algorithms finds the greatest slope for each pixel from a 3x3 patch of heights. Further documented algorithms determine slope using regression techiques, Sharpnack (1969), and vector arithmetic, Ritter (1987), the latter being represented in Figure 5.6.

Aspect is determined by the direction of the maximum slope, for the simplest algorithms this is one of 8 possible directions, corresponding to the neighbourhood pixel positions.

For both slope and aspect rescaling of the available brightness values may be necessary to account for the 0-360° possible aspect values, and the possibility of slopes greater than 100 % (45° or 1:1).

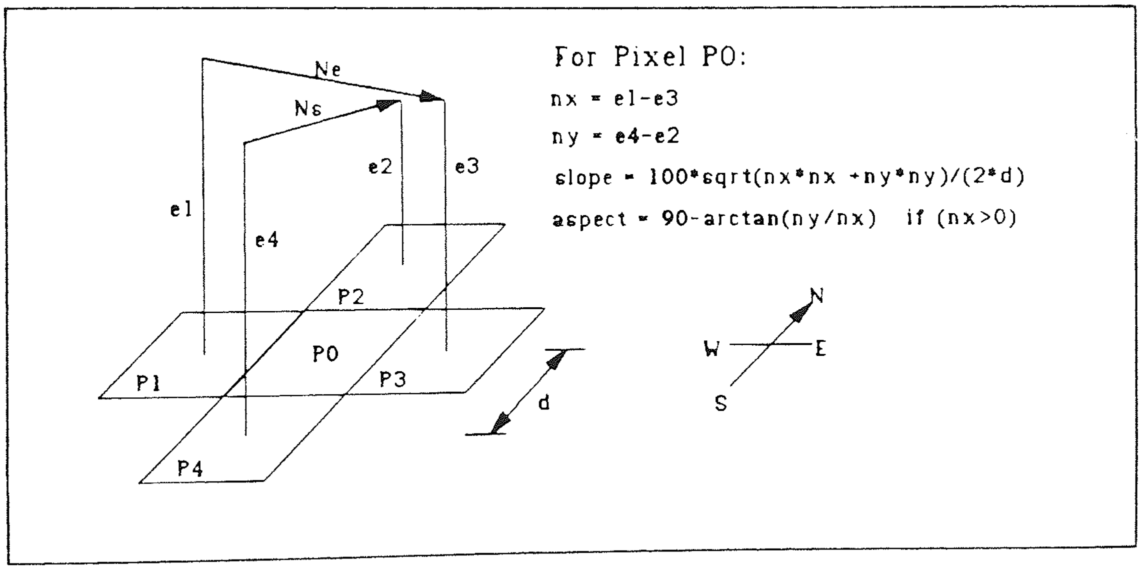


Figure 5.6 Slope and Aspect Calculation, Ritter (1987).

5.4.4 Topographic Partitioning

A point on a digital elevation model can represent a number of hydrologically useful parameters. These include elevation, extreme value (hill top, valley bottom), elevation class, slope class and watershed membership. The DEM has a particular application in watershed analysis, and can be used to derive drainage sub-basin boundaries, depression storage volumes, areas contributing to runoff, and drainage structure.

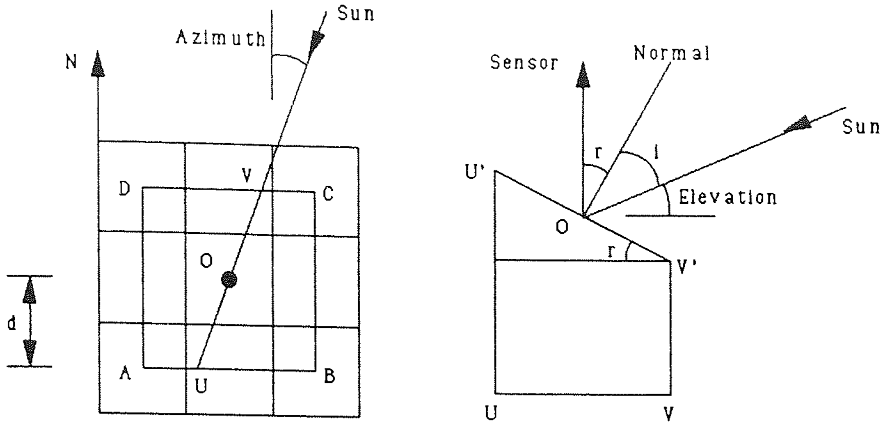
There are a number of methods detailed for partitioning of topography, the majority use simple neighbourhood processes, flagging upwards concave and convex pixels. Band (1986), Collins and Moon (1981) and Jenson and Domingue (1988) detail a number of the processes involved and the potential application of topographic partitioning in topographic and watershed analysis.

5.4.5 Topographic Shading

The production of shaded relief surfaces from digital elevation models relates the slope and aspect of a surface cell to the illumination angle of the light source. This is essentially a graphics rendering problem applied to topographic mapping, and can be found in many computer graphics texts such as Foley and Van Dam (1984).

The main application of topographic shading is in the visual perception of the terrain. Various algorithms for creation of shaded relief topographic products are described in Sprunt (1975), Peucker et al (1975), Batson et al (1976) and Schachter (1979).

A further documented application of shaded relief products is in the quantification and reduction of the topographic effect present on satellite imagery due to solar illumination. This is of particular use in areas of high relief to improve the accuracy of multi-spectral classification algorithms, Justice et al (1981) and Dave and Bernstein (1982).



For Each Pixel (Centre O) :

UU' = ((d-d*tan(Az.) * (Hb-Ha))/(2*d) + Ha

UV = 2*d/cos(Az.)

VV' = ((d-d*tan(Az.) * (Hd-Hc))/(2*d) + Hc

r = arctan((UU'-VV')/UV)

i = 90-El.-r

L = 1/(1 + cos(r)/cos(i))

L = Relative Luminance

Figure 5.7 Relative Luminance under Solar Illumination.

5.4.6 Orthographic Viewing

Given a viewpoint, or direction of viewing, and a co-registered set of digital elevation data and imagery, simulated perspective and isometric views can be created. The production of orthographic views is a geometric problem detailed in standard computer graphics texts.

Combining this form of transformation with image or map data provides an unparalleled cartographic product for perception. The production of 21/2D displays (3D visualisation on 2D display), is detailed by a number of authors including Junkin (1986), Dubayer and Dozier (1986) and Foley (1984) . A particular problem associated with generation of orthographic viewing is that of hidden line or surface removal. Various methods for handling this problem are detailed by Freeman and Loutrel (1962), Loutrel (1970) and Sprunt (1975).

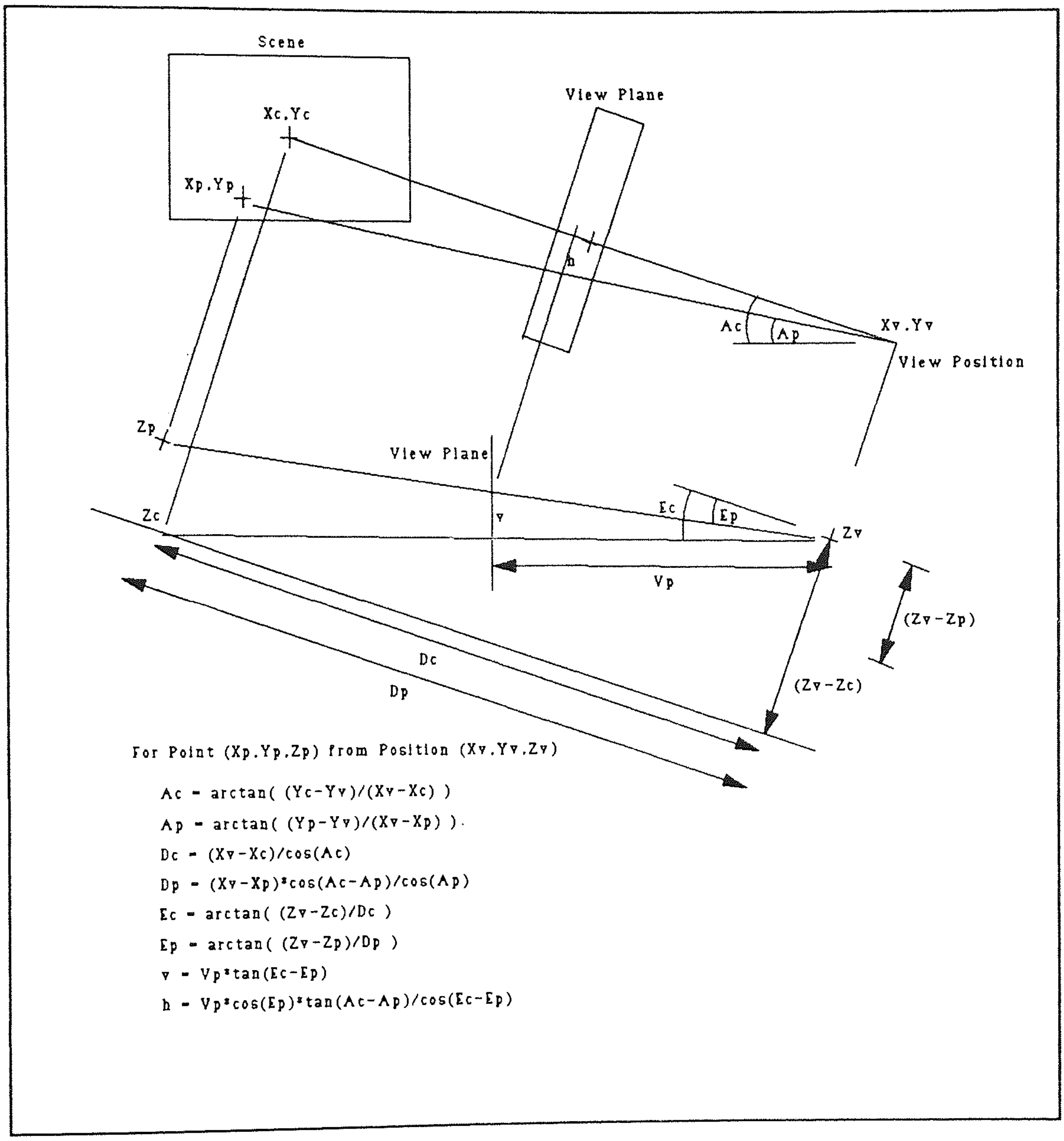


Figure 5.8 Perspective View Geometry.

5.4.7 Intervisibility

The problem of intervisibility between points is easily resolved using a DEM. The visibility of one point from another is determined using a method similar to ray-tracing for hidden line removal detailed by the above authors. An orthographic view generated using ray tracing techniques is simply all points on the surface that are visible from the viewpoint.

A similar application for environmental impact analysis involves the generation of visibility cones from particular points.

6 AIRBORNE VIDEO REMOTE SENSING

6.1 Introduction

Over recent years, the increases in available technology, and decrease in the cost of video technology has introduced video equipment into the domestic market. As the military release their technology, the high quality of equipment becoming available, opens an entirely new field of remote sensing, that of Videography.

Videography, also known as airborne video remote sensing, is the application of video technology to airborne survey. Videography can by no means replace the aerial photographic survey, what it can do is provide a low cost alternative, of particular application in multi-temporal surveys. The low resolution of video sensors is far below that of the grain size of photography and consequently cannot provide the full photogrammetric properties of the aerial photograph. Imagery can however be obtained with a whole range of stereoscopic properties by selecting vertical exaggeration as a function of the number of 1/25th second frames used.

Where videography has its most promising application is in multi-temporal surveys for monitoring purposes. With photographic surveys a significant part of the total survey cost is that of the photographic media, with video technology 270,000 images can be recorded on a three hour tape costing under £5.

In addition to qualitative or interpretative studies, the video imagery can be raster digitised at frame rates using a framegrabber.

Thus quantitative analysis of the data can be undertaken on standard image processing equipment, and the data can easily be geocoded and integrated into information systems.

The advantages of airborne video systems can be summarised as follows:

1. The image recorded can be simultaneously viewed in flight. The aperture and focus can be altered during the flight and therefore there is less chance of failure. In addition imagery of varying scales can be obtained with a zoom lens. The likelihood of obtaining coverage of the area required is significantly improved.

2. The sensitivity of the camera is higher than photographic film and therefore images with smaller bandwidths can be sensed. Video tubes are also designed to operate in a much larger range of light conditions and will perform in low light.

3. The survey can be played back at any stage and its success measured immediately. There is no lengthy wait for return of the data after the survey - analysis can be performed within hours of acquisition.

4. The possibility of direct quantitative analysis on image processing systems via video digitisation. The technology is available and low cost.

5. The overall cost of the survey is low, especially if repetitive coverage is required.

In the field of Water Information Systems and in general engineering applications, airborne video remote sensing fills a gap between the limited resolution of digital satellite imagery and the expense of acquiring high resolution aerial photography. This chapter describes the equipment and techniques used to acquire and process airborne video survey data.

6.2 Video Technology

A major component of an airborne video system, is the video equipment. This consists of video cameras and tape recorders. Analysis and processing of the data then makes use of specific hardware, framegrabbers and decoders. The features of the various components of the system produce the overall sensitivity and resolution of the system and must be examined closely.

6.2.1 The Vidicon Tube

The main component of the video camera is the vidicon tube. This has decreased in size and increased in resolution dramatically since its introduction in 1950.

Light enters the camera through a lens which focuses the image on the light sensitive element of the vidicon tube. This element is covered by a photo-conductive material. The element is scanned by an electron beam which picks up and records the intensity of the light as amplitude on an analogue signal.

The sensitivity of the photo-conductive material controls directly the spatial and spectral resolution of the camera, and varies considerably between manufacturers. The typical vidicon tube fitted to home standard video cameras has a resolution of about 375 lines, and is generally sensitive only in the visible wavelengths.

Perhaps more important are cameras at a level above the home video market with a slightly higher spectral and spatial resolution. Typical of this market are the Model 75 series cameras produced by Insight

Vision Systems Ltd of Malvern. These cameras can be fitted with an extended red vidicon tube, sensitive into the near infrared wavelengths (1100nm), or a tube sensitive in the ultra-violet. These cameras can also have a spatial resolution of up to 625 lines.

Using this type of camera individual spectral bands, as narrow as 5-10 nm, can be recorded using filters. This gives better spectral discrimination than film because optical rather then chemical filters are used. Colour and colour Infrared film relies on the photographic sensitivity of different layers of the film. The sensitivity of these layers overlap resulting in a product which is slightly different from the video image consisting of band sensed in the Green, Red and Infrared wavelengths.

Figure 6.1 shows the sensitivity of a number of different types of vidicon tubes over the visible and near infrared wavelengths.

In addition to cameras with vidicon tubes, cameras are also available that output standard video signals from a CCD (charge coupled device). The CCD is an array of light sensitive cells, each of which outputs independently. These arrays are then interrogated and converted to the standard video signal. Although of higher spatial variation and geometric fidelity, the spectral variation is still limited to the visible wavelengths, unlike the vidicon tubes, and these cameras are more susceptible to vibration. For these reasons the CCD camera is less suitable for airborne surveys.

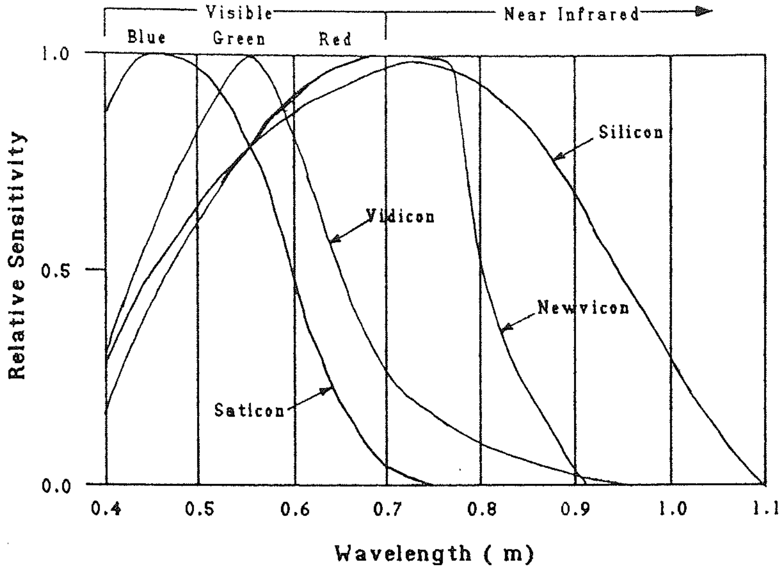


Figure 6.1. Vidicon Tube Sensitivity, Meisner (1985).

6.2.2 Colour Video

For colour cameras an extra degree of complexity is introduced, that of separating the incoming light into its colour components. The majority of colour cameras have three striped colour filters, Red, Green and Blue, in front of a single tube which are used to split the signal into its constituent parts. Each of the filters is orientated in a unique position. The filter then passes the light component which falls on a similarly orientated striped sensor.

Unfortunately these filters absorb a considerable amount of the available light, thus affecting sensing in low light conditions and result in poor definition of thin vertical edges.

With colour cameras using the filters can cause problems, primarily severe diagonal striping on the digitised image. This effect is caused by the reduced resolution available for each component: for example 375 lines per frame gives 125 lines per colour.

An alternative to the filtered single tube camera is the three tube camera. In this more expensive type of camera the incoming light is split into its components using a mirror/prism system. Each component is then sensed by three individual tubes. The infrared wavelengths can also be sensed by altering the position and orientation of the prism and mirrors.

It is this form of camera altered to sense the infrared that has been used to date in videographic surveys. It was developed at the University of Minnesota and is currently being marketed under the name Biovision. Its development, description and applications is more fully described in Meisner (1985) and Maggio (1988). The three tube colour camera is also described as a videographic tool in Vlcek (1983).

An alternative approach was described by Nixon et al (1984). This was a direct comparison between Colour Infrared photography and video in crop management. The system used was a single B/W video camera with different optical filters on subsequent overpasses. Later work by the same authors, Nixon et al (1985) and with Everitt (1985), detailed a multiple camera system, simultaneously recording 3 or 4 bands of data on separate recorders.

The encoding of the separate signals is inherent in all colour video systems and produces a composite colour video signal. This encoding maintains compatibility with BW equipment and results in the degradation of the individual components. If quantitative analysis is required then a multiple camera system should be used. Qualitative work can normally be adequately carried out on a single camera system, the advantage being the lower cost of the system.

6.2.3 Lenses

The standard 2/3 inch format vidicon tube has a picture size of 8.8 x 6.6 mm. In comparison with 35mm photographic equipment this means that the focal lengths of lenses are approximately 4 times as long. Thus 8.5, 12.5, and 25mm lenses give the comparative 34, 50 and 75mm lenses in 35mm photographic equipment.

Video lenses can therefore be classed as:

8.5 mm - Wide Angle
12.5 mm - Normal
25.0 mm - Telephoto

The selection of lenses is therefore an important factor in selection of survey parameters as it controls both flying height and ground coverage.

The lenses also have a distinct effect on the geometry of the system. Video cameras are not specifically designed with geometric fidelity in mind. Consequently there are numerous geometric errors which have to be corrected before subsequent processing. The majority of errors are introduced by lens effects and consequently can be kept to a minimum by using longer or normal focal lengths. The major distortion, the barrel effect, is more distinct at shorter focal lengths. This distortion can be mathematically modelled and consequently removed with higher order geometric correction processes.

6.2.4 Spectral Filters

To sense in specific wavebands using extended red vidicon tubes requires the combination of a number of filters. Filters typically transmit light within a given range of wavelength. Care must also be taken to exclude unwanted wavelengths, for example a visible red filter used with an extended red vidicon tube will sense from red to the near infrared. Table 6.1 shows certain combinations of waveband transmitting and blocking filters that can be used for sensing in specific wavebands.

Waveband	Filters	Type-Make
Blue	UV Blocking Blue Transmitting IR Blocking	Kodak Wratten 47 / 47B
Green	UV Blocking Green Transmitting IR Blocking	Kodak Wratten 58 / 61
Red	UV Blocking Red Transmitting IR Blocking	Kodak Wratten 25 / 29
Near IR	Visible Blocking	Kodak Wratten 89B

Table 6.1. Spectral Filters for Airborne Video.

6.2.5 Thermal Video

In addition to the visible and infrared wavelengths, certain video cameras can sense in thermal wavelengths. These cameras use Germanium or Zinc Selenide lenses and can sense at wavelengths up to 22 microns. Glass absorbs thermal wavelengths and therefore these cameras have to be mounted with direct access to the atmosphere.

The difference in cost between thermal video and conventional line-scan methods is large, making thermal video particularly attractive. Special consideration for thermal radiation must be considered in design of the aircraft mount. This makes the airborne thermal survey considerably more expensive and therefore beyond the scope of this study.

6.2.6 The Video Tape Recorder

The requirements of a video tape recorder for use in airborne surveys is that it must be fairly light, robust and be able to powered by either batteries or the internal electric system of the plane.

The tape recorders available at present fall into four major categories, Umatic, Betamax, VHS and VHS-C.

The Umatic tape recorder is the system used in industrial applications. This uses a 3/4" tape and is therefore generally bulkier and heavier. With the recent advances in this field the readily available low cost 1/2" format tapes are now of comparable quality and are also portable, making them the ideal choice for this form of survey. Between the 1/2" tapes there is little to choose, the main factor being the availability, some 75% of the video equipment sold today is VHS format.

The video tape recorder is to some extent the weak link in the system having a limited recording resolution of about 325 lines. This may well be an area of significant improvement over the next few years, with the introduction of affordable digital video tape recording systems.

6.2.7 Video Signals

There exists a number of video 'standard' signals. In the UK this standard is the CCIR standard, which most cameras and video tape recorders use. This analogue signal contains information on the Red, Green and Blue or total luminance levels and the synchronisation for the interlacing of the odd and even fields in the frame.

In colour systems the separate signals received from the vidicon are encoded to give the composite video signal. The most common form of encoding of the signals in the UK is the PAL standard.

The reason for this encoding is to maintain compatibility with monochrome equipment. In this encoding a standard transformation is applied known as the YIQ transformation. This transformation creates a luminance signal (Y), equivalent to a monochrome signal, from the weighted sums of the RGB signal. Colour is then expressed by two further signals luminance minus Red (I) and luminance minus blue (Q). The I and Q are then combined to give a chrominance signal which is frequency modulated and combined with the luminance.

Unfortunately the encoding of the separate components into a composite signal reduces the amplitude of each signal, consequently a large amount of the spectral amplitude is lost.

6.2.8 Video Digitisation

The hardware required for digitising video signals is often referred to as a Framegrabber. The introduction of this hardware has enabled airborne video survey data to be digitally enhanced and processed, in much the same way as digitally recorded data.

The framegrabber generally accepts a number of inputs. These include monochrome and synchronisation composite signal and Red, Green, Blue and Synchronisation separate signals for a full colour framegrab. Colour digitisation, from cameras or tape recorders, requires a PAL decoder to split the composite signal into its components.

The image or frame consists of two interlaced fields; one field being the odd, the other the even lines of the frame. Each set of lines is drawn separately, with the speed of the re-draw controlled by the synchronisation of the signal. The CCIR standard operating at 50 Hz redraws a field every 1/50th of a second. Thus 25 images or frames are digitised every second, similar to the interlacing of a standard television display.

The framegrabber card used in this study, digitised the incoming signal to produce an array of 768 pixels per line with 576 lines. This is well above the resolution of both the camera and the video tape recorder.

6.3 Survey Systems

A number of airborne video survey systems have been investigated. The difference between these systems lies in the cameras and recording equipment, and in the method of obtaining multi-spectral image data.

The first system described was used in the initial surveys undertaken to assess the feasibility of airborne video systems for various applications. Chapter 8 - the Taw-Torridge water quality survey, describes the application of this system in a water quality survey.

Two viable alternatives to the initial system are proposed. These systems are specifically designed to overcome the problems encountered on the initial surveys.

6.3.1 Aircraft

The aircraft used for all the initial surveys was a Cessna 172 based at Coventry airport. The Cessna 172 is an extremely robust, single engined, high winged aircraft, that has become the workhorse of numerous flying schools and clubs.

The aircraft required modification before it could be used for survey work, to provide a clear vertical view of the ground. Any modification work carried out had to meet with the approval of the Civil Aviation Authority.

The Institute of Hydrology had recently undertaken some survey work with 35mm cameras, Blyth (1978). For this survey IoH used a Cessna 172 door modified to provide a special housing that provided an area outside the normal fuselage of the plane, accessible from within the fuselage. The cameras were mounted on a free swinging gantry that helped in maintaining a near vertical look angle. For airborne video survey work the gantry had to be slightly modified to accommodate video cameras and equipment.

Figure 6.2 shows the layout of the aircraft, modified door and vertical viewing system.

The door of a Cessna 172 is held on with two split pins and a lock. The process of removal and fitting the door is therefore relatively simple.

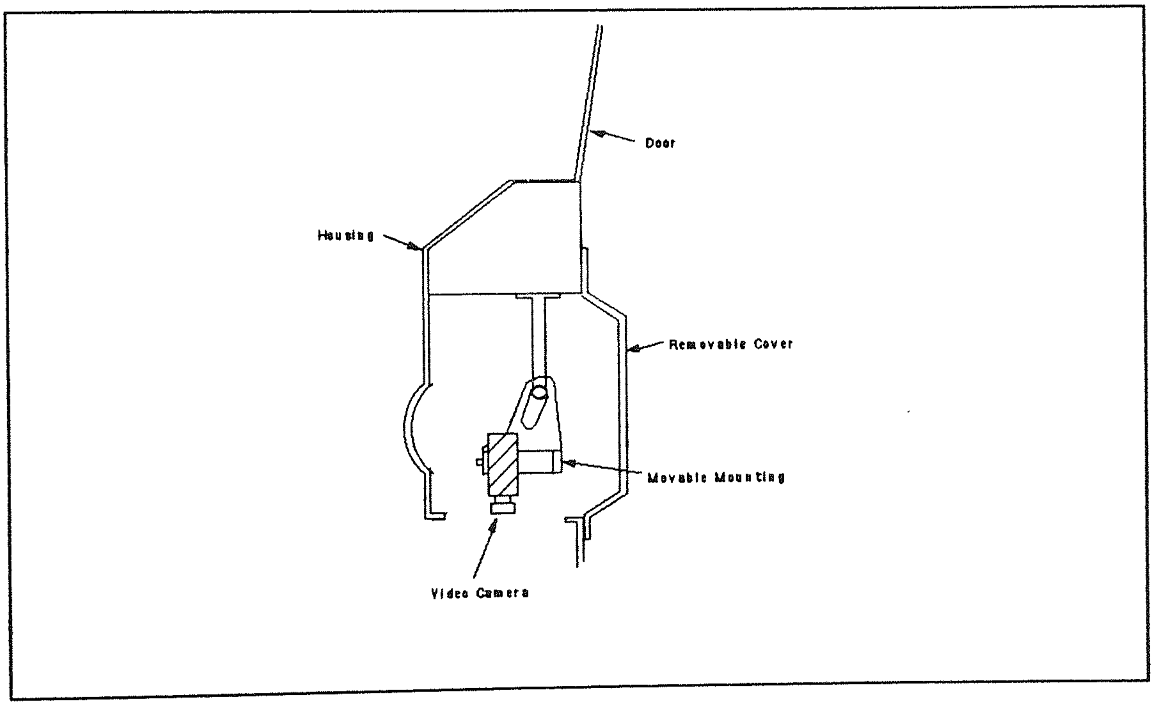


Figure 6.2. Aircraft Door and Video Camera Layout.

During the flight the survey is recorded on tape recorders and simultaneously viewed on monitor capable of switching to any recorded signal. This monitor is placed towards the front of the cockpit so that both the pilot and survey co-ordinator can view the scene. This allows the pilot to correct to the specified flight path and allows the co-ordinator to ensure the survey information is sufficient. The real time viewing of the recorded survey, and ability to alter lens and camera settings, is regarded as one of the major advantages of airborne video over photographic survey.

6.3.2 Video Cameras and Video Tape Recorders

The system used in the initial surveys is detailed in Figure 6.3. Two video cameras and two video tape recorders were used; a single tube, domestic quality, colour camera to record the visible wavelengths; and a model 75 series Insight Vision BW camera fitted with an visible light blocking filter to record the near infrared wavelengths. Both signals were recorded on portable VHS tape recorders.

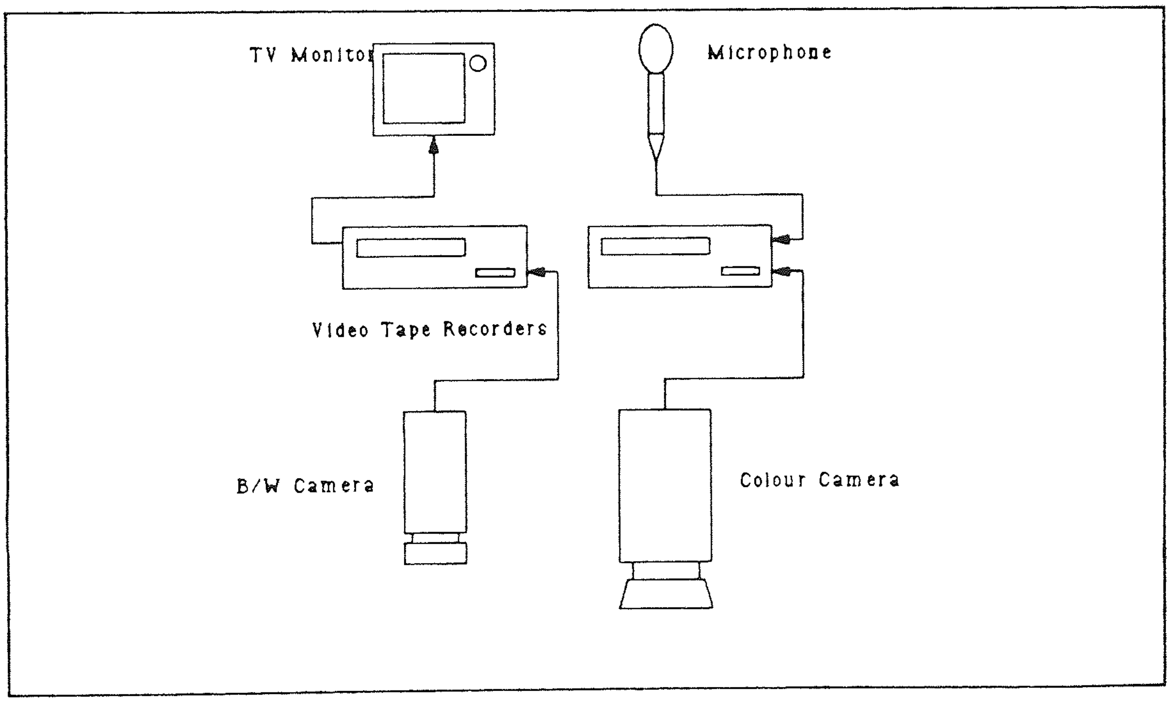


Figure 6.3. Airborne Video Survey System.

The results of this survey suggested two possible improvements. Both were aimed at removing the necessity for using single tube colour cameras and therefore improve the radiometric quality of the recorded signal.

Recording multispectral imagery requires as many separate sensing and recording systems as spectral bands of data required. For example, the main four wavebands of interest are in the Blue, Green, Red and Near Infrared wavelengths. This can be achieved in two ways:

1. Arrays of cameras can be used. For each waveband of sensing there is a separate camera and tape recorder. Each camera need only be a BW camera filtered to sense in the required wavelengths. The requirements for cameras and recorders and power required may be excessive, but the subsequent processing is significantly easier. Also radiometric fidelity is more certain, with no encoding of the signal prior to recording, and no decoding of the signal prior to digitisation.

   The timing of the cameras can be driven from one source such that for each camera every interlaced field and frame is synchronous. The process involved in constructing the colour composites is simply to find the corresponding frame on each recording.

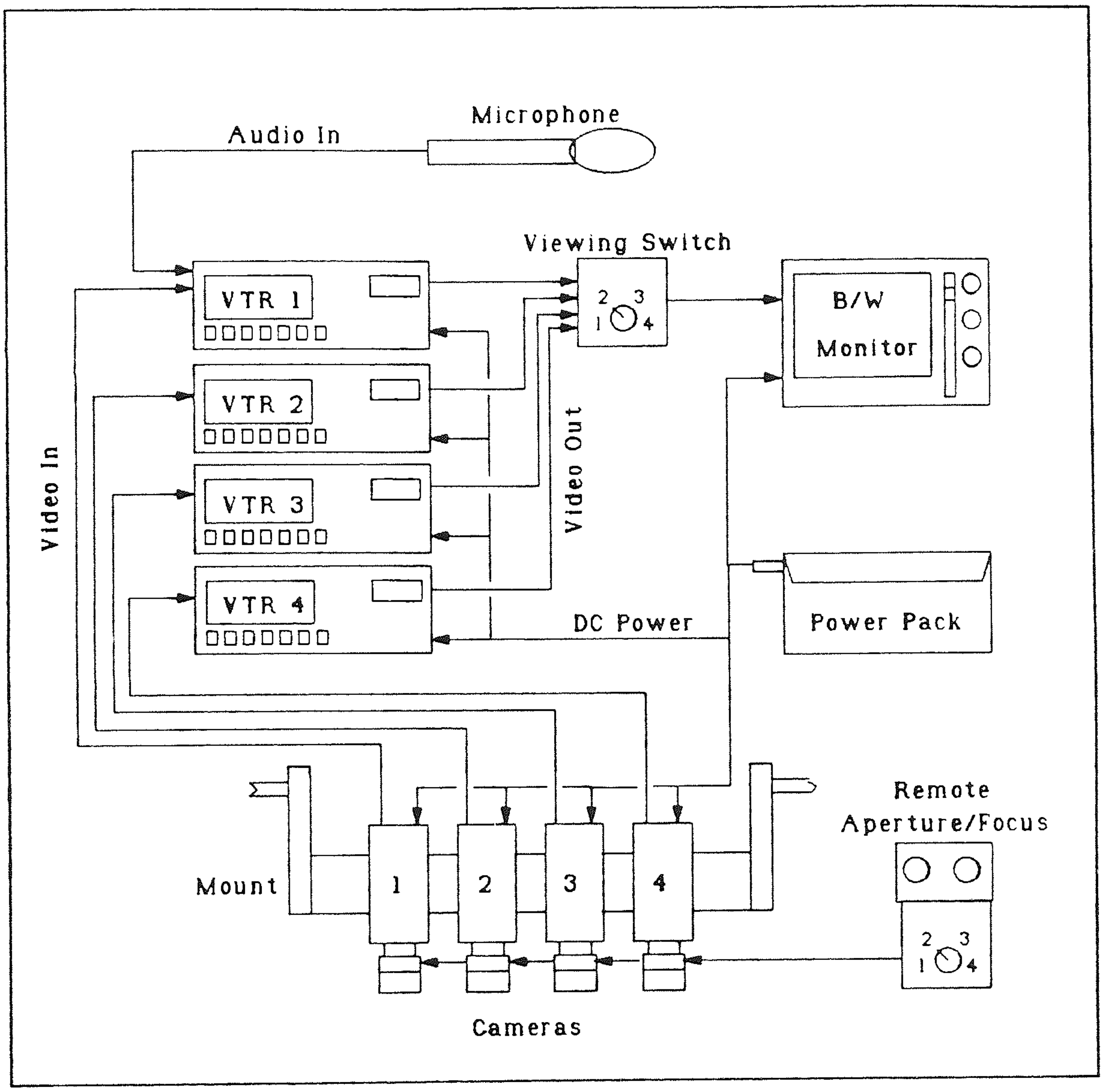


Figure 6.4. Proposed Airborne Video Survey System.

2. A single BW camera recording on a single tape recorder is an alternative to the multi-camera system. Individual wavebands of sensing are acquired using a moving filter system. Given that a frame is recorded every 1/25th of a second, the filter must be stationary for at least that time, must move to the next filter within 1/25th second and must cycle through all filters in the time taken for the aircraft to move a significant distance.

The design of such a system is therefore fairly complicated. The advantage of this system is that as many wavebands can be recorded, as can fit into the constraint of significant scene movement in terms of flying height, field of view and aircraft speed.

The processing of the recorded data then relies on being able to extract individual frames from the tape. Theoretically this system has many advantages over the previous systems, however the design and development of filter movement and efficient digitisation systems could prove prohibitively expensive.

Irrespective of survey system used certain aspects of the system require further considerations. The majority of cameras currently available,have a number of in-built features which, if the data is to be used for quantitative analysis, have to be disabled.

If a video mosaic is required then the auto-gradient, auto-contrast control must be disabled. This systems work by analysing the scene to determine the ambient brightness. Auto-gain and offset routines, built into the circuitry, or alteration of the aperture of the lens, adjust the brightness of the resultant image. There is also generally a lag in the adjustment process.

Some cameras also have automatic edge enhancement and a variety of other automatic features. The general rule for all these features is to disable everything and give the control of the sensing to the operator.

If real time viewing of the scene is to be used to its full advantage, then some means of controlling the lens aperture and focus from within the cockpit must be provided. It was found during the initial surveys that adjustment of lens aperture and focus should be carried out over the survey area, and not on the ground prior to take off. This removes possible errors and movement of controls caused by vibration during take-off.

Some form of calibration for variable lighting is also advantageous, especially if multi-temporal surveys are planned, although this can be accounted for using digital image processing techniques.

For location of flight paths, after the survey, a 35mm photographic camera, fitted with a wide angle lens, is invaluable. The times and locations of the photographs taken should be recorded on the soundtrack of the video and on a flight log.

6.3.3 Flight Planning

The logistics of flight planning involve consideration of a number of factors. These include flying height, flying speed, lenses, ground coverage and flight paths. The selection of camera and recording systems has already been discussed in the previous section.

Flying height, flying speed, selection of lenses, and ground coverage are all interrelated. Sufficient ground coverage can normally be achieved with proper selection of lenses. Table 6.2 can be used to assess the suitability of available lenses for different flying heights and required ground swaths.

Ground Width (m)	Pixel Size (m)	Lens 6.0 mm	Lens 8.5 mm	Lens 12.5 mm	Lens 25 mm
100	0.2	223	317	465	930
200	0.4	446	634	932	1861
250	0.5	556	793	1165	2326
500	1.0	1119	1586	2329	4652
1000	2.0	2237	3171	4658	9303
2000	4.0	4474	6343	9316	18606
2500	5.0	5593	7929	11645	23258
5000	10.0	11186	15857	23291	46516

Table 6.2. Flying Height (ft) for Several Video Camera Lenses.

As can be seen from this table a number of the options are not available to planes without cabin compression and only a few options lie in the preferred range of heights for light aircraft (1000-3000ft).

The flying speed and height also control the movement of the scene across the vidicon tube plate. The problem is whether or not there is a significant movement of the scene sensed, in the field interlace time - 1/25th of a second.

The image is formed on a 8.8 x 6.6mm plate. This is then digitised to a 768 x 576 image in two interlaced fields with the fields aligned along the line of flight. For the movement to be significant the scene must have changed by 1 pixel that is 8.8/768 = 0.0115mm. The distance moved in 1/25th of a second is then determined from the aircraft velocity.

As an example with the aircraft flying at 175 kmph at 3000 ft (914.4 m) with a 12.5 mm lens.

Lens Field of View = 2 x arctan(4.4/12.5) = 38.78 deg.

Distance in 1/25th sec. = 175000/(60 x 60 x 25) = 1.94 m.

Corresponding distance on tube image = 1944 x 12.5/914400 = 0.02658 mm

Equivalent to 2.3 pixels therefore the interlaced fields will be out of registration by 2.3 pixels.

This calculation can be inverted to give the maximum velocity for a given lens and flying height.

For the same system

1 Pixel maximum difference - 0.0115mm on image plate.

Distance on ground = 0.0115 x 914400/12.5 = 0.84 m.

Maximum aircraft velocity = 60 x 60 x 25 x 0.84 = 75.6 kmph which is generally too slow to fly!

Alternatively, given the minimum flying velocity the flying height can be determined.

The design of overlapping flight paths to provide total coverage of the study area is described in any airborne survey text. Consideration of overlap along the flight path is not required due to the nature of the data. In terms of operational acquisition of data this requires the identification of landmarks along each flight path, and consideration of the turning circle of the aircraft.

6.4 Integration with Information Systems

Digital video imagery can be directly integrated into an information system. The conversion of video analogue imagery from video tapes to geocoded image data involves decoding of the signal, digitisation of the separate inputs, co-registration of multi-spectral data, storage of frames, and geographic registration of frames onto the total database coordinate system.

The processes involved are operationally intensive in terms of processor power and storage. Given the limitations of unenhanced micro-computers the development of the system was restricted to small frame sizes and limited resolution. A operational system for rapid integration of video imagery into a cartographic database becomes a possibility with the addition of dedicated processors (hardware geometric warpers), programmable Digital Signal Processors or Transputers.

Once integrated onto a common geographic base the image can be merged with other data, digitally enhanced, and processed to extract information at a scale comparable with aerial photography, at a fraction of the cost. Discussion with various interested authorities have highlighted a number of potential applications of this imagery. The applications considered include :

1. Urban land use change detection, to co-ordinate teams of surveyors updating OS Topographic Maps.

2. Automatic classification of urban areas to determine location and extent of impermeable-permeable areas for use in runoff calculations and urban drainage design systems.

3. Agricultural land use classification, for crop monitoring, vegetation stress analysis and identification of groundwater pollutant sources.

4. Water quality mapping, determination of turbidity, suspended sediment concentration, chlorophyll-a concentrations and algal blooms in river, estuarine and coastal waters.

6.4.1 Decoding and Digitisation

The first stage in the processing was to digitise the imagery stored on the video tape. With colour video recording the tape is played back and digitised through a PAL decoder that splits the composite colour video signal into its Red, Green and Blue components. This produces 24 bit colour imagery directly. The encoding of the signal reduces the dimensionality of each colour component; each component when decoded and digitised has varying amounts of noise which must subsequently be removed.

With multiple camera video surveys each composite scene must be reconstructed from multiple tape recorders. If the recorders can be synchronised then composite digitisation can be performed. If a single camera, multiple filter, system is used, then the composite image must be reconstructed from consecutive frames. In both cases simple horizontal shift is generally all that is required to co-register spectral bands for a single scene.

6.4.2 Preprocessing

Preprocessing of the data is necessary to remove various radiometric errors introduced by decoding, digitisation, interlacing effects and noise.

The framegrabbing process digitises the odd or even fields of the frame every 1/25th of a second. If the flying speed and height have not been taken into account, the scene may have moved by a significant amount in the interlace time and the fields may not match. This can be rectified by using only one of the interlaced fields and replicating the second, a repeat field grab. Alternatively every second pixel on every second line may be extracted, producing an image exactly half the size of the original.

Various image processing techniques exist to remove noise from images. This can simply be neighbourhood convolution filters, replacing each pixel by the mean brightness of itself and its eight neighbours; or it may involve complex filtering methods such as fast fourier transforms.

These two methods attempt to reduce noise whilst preserving edges and linear features. Both provided comparable results, with fourier filtering more effective at reducing specifically orientated noise. There was a significant difference in timings; convolution filters taking of the order of 20 seconds, fourier methods 40 minutes. There are a number of available fourier filtering hardware products, without these the time taken to perform fourier filtering is prohibitive.

6.4.3 Storage

Each digitised frame is chosen such that there is a minimal overlap. This is done by framegrabbing continually until a recognisable feature moves from one side of the frame to the other. This means that every 2 or 3 seconds a significant image is produced and saved.

During a test programme, approximately one hour was flown at a height of 3000ft. At 3000ft a new frame overlapping 10-20% with the previous frame is acquired every 2 seconds. In one hour 1800 images would be produced occupying some 1.4GB of 512x512x24 bit storage. The only systems capable of providing this amount of storage on micro-computers, in a readily accessible format are optical storage devices. The vast amounts of information necessitates compromise if processing is to be completed in the time available, or accelerated processing and automated processing of the data.

6.4.4 Registration to Cartographic Database

To correct image data to a spatial database co-ordinate system involves the use of ground control points. These are points identified on each image for which the map co-ordinates are known. These points are used to derive a transformation matrix by solution of simultaneous equations by least squares methods. The process of transformation of the image data is known as geocoding, geometric registration or warping to ground control points. Were this method to be applied to the large quantities of video data produced the overall time for production of a geocoded video-mosaic might well be beyond an operational time scale.

To increase the production process the control of the geometric correction would have to operate in near real time. An operational system to provide geocoding of frames, and construction of mosaics has been developed on a small scale. Enlargement to full scale would involve the addition of computational accelerators, hardware warpers, and larger framestores.

The system developed works on digitised colour video frames reduced to 128x128 pixels. The system makes use of a digital map database, acquired from a mapping agency, or produced using the methods described in Chapters 4 and 7. The window on the database co-ordinate system is defined and the digital map data recalled onto the non-destructive overlay planes of the framestore.

The video frame is read into memory and becomes a 24 bit colour live cursor controlled by the mouse. Moving the mouse moves the cursor around the display. The right mouse button provides a link to a routine that alters the degree of rotation and scaling of the frame. The image cursor is then moved under the digital map features on the overlay planes until the image matches the map and any image data on the display. Where consecutive frames overlap procedures are used to ensure a seamless mosaic.

Figure 6.5 shows a typical mosaiced area, 1km in size, constructed from 200x200 m video frames, with the feature data extracted from the map overlayed.

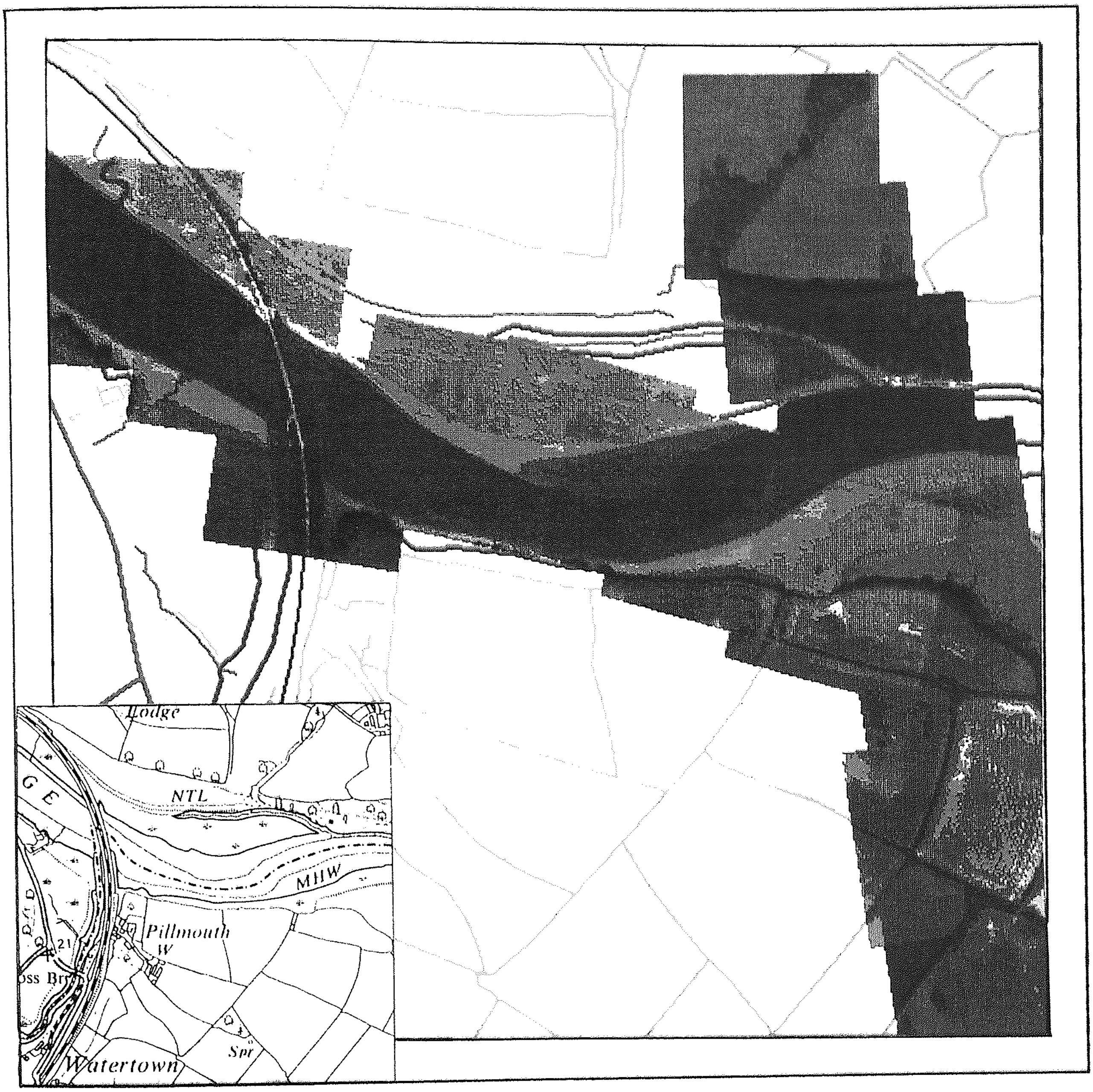


Figure 6.5 Typical Mosaiced 1km Video Map.

7 THE WATER INFORMATION SYSTEM

7.1 Introduction

The water information system is a spatial information system designed for applications in water resources management. It is therefore geared to processing of data for one of the most fundamental hydrological units - the river basin. The scale of the river basin is such that the system can be designed around the latest microcomputer technology. Its low cost enables its use as a portable project based system for resource mapping and monitoring.

Water management policies draw upon a wide range of data in support of decisions. These data include the monitoring and prediction of hydrological parameters such as water quality and runoff; and the mapping of features such as land use. The river basin information and management system can therefore be seen as a overall technology, involving several integrated systems, each dedicated to a particular task. Certain aspects of the technologies involved have already been described in previous chapters.

Figure 7.1 shows the overall concept of the proposed water information management system. It includes the reception of meteorological image and telemetered data transmissions, providing data for monitoring and modelling of rainfall, runoff, flooding, erosion and water quality; the integration of data from loggers and survey equipment, large scale digital satellite imagery, videographic imagery and digital maps; and the production of topographic and thematic databases.

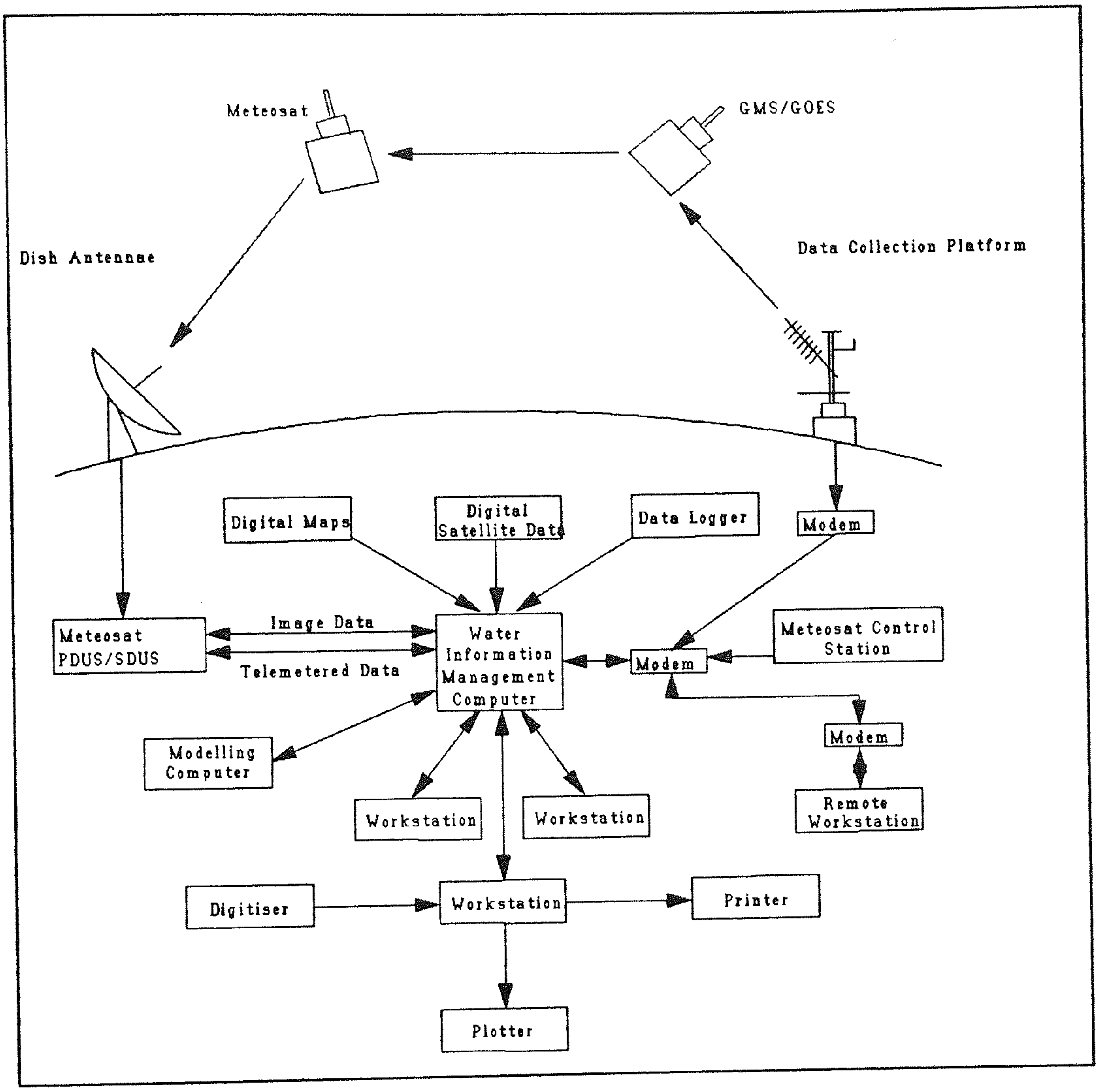


Figure 7.1. The Water Information System.

7.2 Software Engineering

The majority of software engineering design principles are geared towards large products developed by a number of programmers. In this environment the system is generally broken down to single entry - single exit routines, and each can be designed in parallel by a group of programmers.

The disadvantage of this is that the total system overview is lost to all but a few programmers. For this reason the level of fragmentation should not be too excessive.

7.2.1 System Requirements

To commence the design process the overall requirements of the systems performance need to be examined. This has been discussed in chapter 2 and has resulted from a review of current information systems and the typical requirements of water engineers.

Figure 7.2 details the various modules and functions in the water information system, and is a schematic representation of the requirements. The total system can be divided into a number of sub-systems.

Telemetry system
Image system
Mapping System
Analysis System

Various components of these sub-systems have been studied in greater detail in Chapter 3, namely the software and hardware technology involved and the design of the man machine interface.

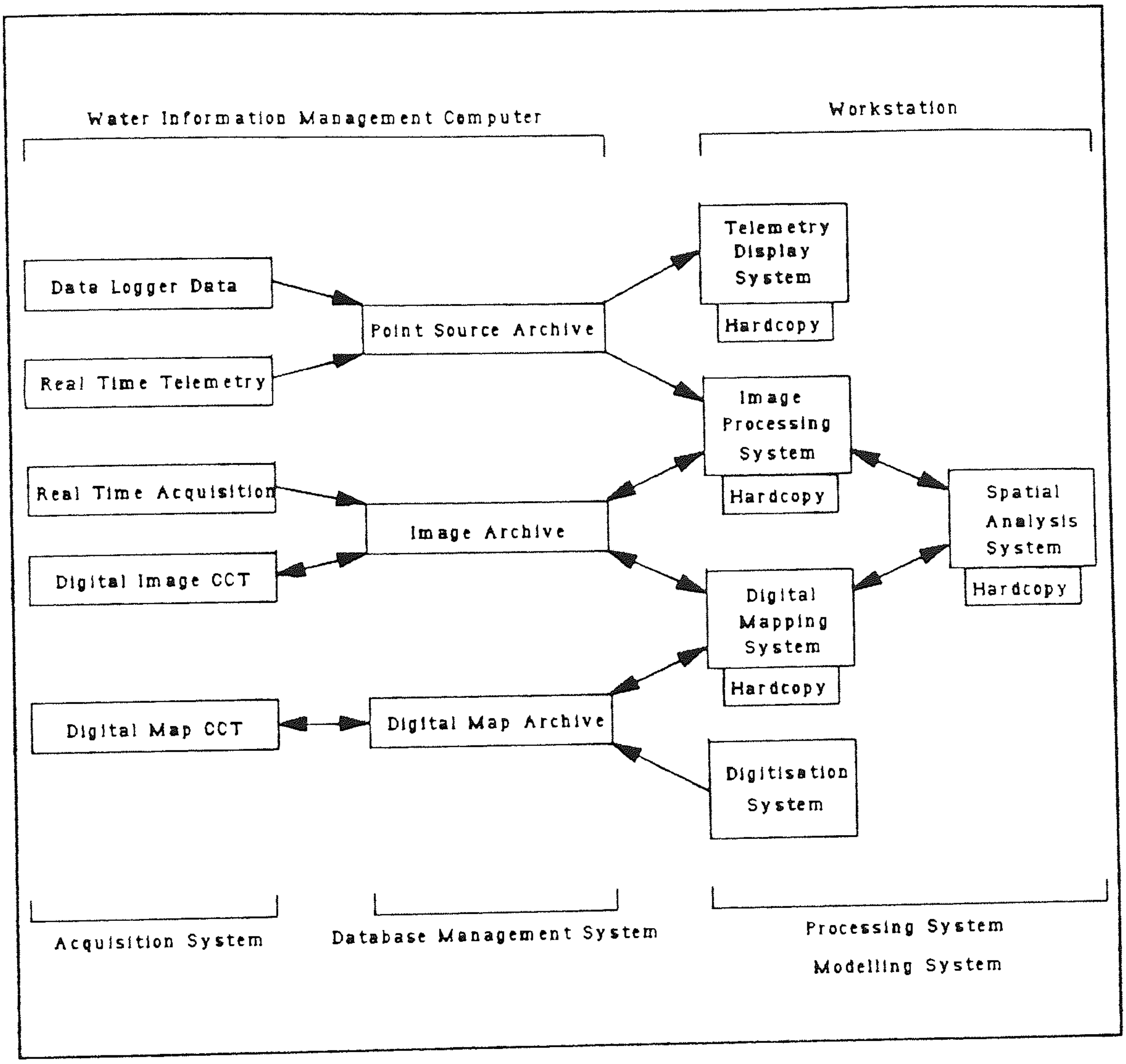


Figure 7.2. Software Requirement Specification for the Water Information System.

7.2.1.1 Telemetry System

Point source, time related data can be integrated into the system from Data Collection Platforms (DCPs) via satellite telemetry. This is described more fully in Chapter 3. The requirements of the software system to manage the telemetered data must perform the following functions.

1. Scheduling of data to be received
2. Real time acquisition of data relayed
3. Update and maintenance of historical/rolling archive
4. Data display
5. Time Series and Spatial analysis of data

This system requires a satellite dish antennae and receiving system. The real time system requires a certain amount of system level programming to manage acquisition of the data and disk update during normal operation of the host machine. The complexity of this level of programming is dependent on the host operating system.

7.2.1.2 Image System

The image sub-system includes a wide range of functions for the real time acquisition of image data from meteorological satellites, the integration of digital remotely sensed imagery supplied on CCT, the acquisition of video imagery from tape recorders or cameras, and the processing of all image data to produce thematic map products. Functions include:

1. Real time system
   a). Scheduling of data to be received
   b). Real time acquisition of image data
   c). Update and maintenance of historical/rolling archive
2. Tape reader system
3. Video digitisation system
4. Image Processing system
   a). Operating system interface
   b). Update and maintenance of image store
   c). Display of high resolution image data
   d). Calibration and correction of data
   e). Enhancement of image data
   f). Display of image data statistics

g). Classification of image data

h). Band manipulation and transformation

i). Overlaying and annotation

j). Hardcopy

Real time acquisition of image data requires a dedicated ground receiving system. The components of a tape reader, video digitisation and image processing system are represented in Tables 7.1, 7.2 and 7.3. The functionality of these components represents the minimum requirements for the processing of remotely sensed data.

TAPE SUPPORT MODULE	
Analyse Tape	Files Records Dump Record ASCII/EBCDIC
Read Tape	Read RAE Read CCRS Read EROS General Purpose
Write Tape	Write RAE Write CCRS Write EROS General Purpose

Table 7.1. Tape Support Module.

VIDEO DIGITISATION MODULE	
Preview	Monochrome Colour
Grab Frame	Monochrome Colour
Grab Reference	Monochrome Colour
Shading Correction	
Image Enhancement	Density Slice Contrast Enhancement Convolution Filters
Image Utilities	

Table 7.2. Video Digitisation Module.

IMAGE PROCESSING MODULE

DOS Interface	Set Drive & Directory	
	Find a File	
	Delete a File	
	Create a Directory	
	Remove a Directory	
	Print Logfile	
Utilities	Define an Extract	
	Clear Extract	
	Save/Restore an Image	
	Mask an Area	
	Copy an Image	
	Display Component/Overlay	
Corrections	Destripe	
	Geometric Correction	
	Line Drop-Out Removal	
Enhancements	Contrast Enhancement	Remove Stretch
		Auto-Linear (x4)
		Auto-Gaussian
		Auto-Equalise
		Manual Piecewise
		Transform Image
		Save/Restore Stretch
	Low Pass Filters	Meanal
		Median
		Modal
		User Defined Kernel
	High Pass Filters	Laplacian
		Horizontal Edge
		Vertical Edge
		Roberts Edge
		Sobel Edge
		User Defined Kernel
Statistics	Histogram	
	Scatter Plot	
	Transect	
	Read Pixel	
	Measures (Mn, SD, etc.)	
Classifications	Density Slice	Define Slice
		Switch Component
		Remove Slice
		Transform Image
		Save/Restore Slice
	Define Training Areas	
	Derive Class Statistics	
	Box Classification	
	Minimum Distance	
	Maximum Likelihood	
	Class Statistics	Histogram
		Scatter Plot
		Measures (Mn, SD, etc)
	Class Areas	

IMAGE PROCESSING MODULE (Continued.)	
Transformations	Band Arithmetic Principal Components Colour Space
Overlays	Draw Lines, Boxes Label Image
Hardcopy	Print False Colour Print Single Component Print Overlay Print Density Slice

Table 7.3. Image Processing Module.

Given the delay in acquisition of high resolution satellite data, other methods must be used to calibrate real time data. These may include use of telemetered DCP data, or alternatively airborne video remotely sensed imagery provides the quickest method of acquiring high resolution image data. This has the added advantage of easy integration to the system using video digitisation technology.

7.2.1.3 Mapping System

The mapping system provides the necessary links to topographic and thematic data, enabling image and map data to be integrated within the system. The background to the design and development of the digital mapping system, as part of the total spatial information technology, is described in Chapter 2. Digital Map data is available in numerous formats, generally supplied on computer compatible tapes (CCTs), and utilities must exist to transfer data between internal and external formats. A further source of digital map production is the raster digitisation of cartographic products within the system, as described in Chapter 4.

The system must have adequate facilities to update the digital map data using image or model data, and be able to handle cartographic data of varying scales, projections and units. Table 7.4 details some of the functionality of the digital mapping system. A further description of the system and its application is included in Chapter 8.

DIGITAL MAPPING MODULE		
Feature Codes	Select Code Tree	
	Edit Code Tree	
	Create Code Tree	
	Print Code Tree	
	Define Symbol Table	
Feature Data	Select Feature File	
	Define Viewport	
	Define Features	
	Recall Features	
Elevation Data	Select Elevation File	
	Define Viewport	
	Define Datum/Units	
	Define Elevation Feature	
	Recall Elevation Feature	
Edit Feature	Select Feature	All
		Several
		More/Less
	Move Feature	
	Copy Feature	
	Delete Feature	
	Un-delete Features	
	Clear Deletions	
	Edit Feature Vertices	Select Vertex
		Move Vertex
		Add Vertex
		Delete Vertex
	Edit Feature Attributes	
Utilities	Concatenate Files	
	Generalise Linear Features	
	Import Data	
	Export Data	

Table 7.4. Digital Mapping Module.

In the field of hydrology and water resources a large number of applications involve integration with models, such as runoff, soil erosion and water quality models. These invariable use some measure of topography (elevation, slope, aspect), and therefore inclusion of surface modelling functions, such as interpolation and viewing transformations, are a necessity. Table 7.5 shows the functionality of the surface modelling system.

SURFACE MODELLING MODULE		
Interpolation	Random Input	Triangulation/Bi-linear
		Weighted Average
		Least Squares
		Multiquadric
	Grid Input	Bi-linear
		Bi-cubic
Viewing Transform	Select Surface	
	Select Image	
	Define View Direction	
	Perspective	
	Isometric	
Derivations	Slope	
	Aspect	
	Shaded Relief	
	Intervisibility	
	Watershed	
	Stream Channel	

Table 7.5. The Surface Modelling System.

7.2.1.4 Spatial Analysis System

The spatial analysis system provides a means of relating spatial databases, to examine their interrelationships, and produce thematic products. This sort of system enables raster maps of various features to be created from boundary information, and must be able to incorporate all spatially related geographic information, including satellite image and telemetered data.

The background to this component of the spatial information system has been presented in Chapter 2. Table 7.6 shows some of the functionality of the spatial analysis system, used in development of the applications described in Chapter 8.

SPATIAL ANALYSIS MODULE		
Class Table	Select Class Table	
	Edit Class Table	
	Create Class Table	
	Create Key Display	
	Print Class Table	
Create Class Map	Select Feature File	
	Define Viewport	
	Recall Boundary Data	
	Fill Bounded Areas	
Edit Class Map	Remove Boundaries	Line Follow
		Convolution
	Paint	Class Value
		Surface Value
		Smooth
Proximity	Select Feature(s)	All
		Several
		More/Less
	Proximity Surface	Variable Weighting
		Constant Weighting
	Voronoi Diagram	
	Proximity Corridors	Variable Distance
		Constant Distance
Relate Map Classes	Select Map	
	Define Operations	
	Define Map Combinations	
	Generate Result	
Measurement	Class Area	
	Linear Feature Length	
Utilities	Define Extract	
	Clear Extract	
	Save/Restore Encode/Decode Map	

Table 7.6. The Spatial Analysis System.

7.3 The Man Machine Interface

In design of the man machine interface several interactive devices and error management procedures were developed. These are presented in the following sections. It should be noted that at this level of programming a certain amount of device dependency is necessary.

7.3.1 Interactive Devices

For the purpose of design of a water information system, several interactive devices and techniques were developed to simulate the locator, pick, valuator and button using an optical mouse and a keyboard.

The interaction system developed in this study was different from a large number of systems in that two graphics displays were used. The image or graphics display was controlled by an imaging board as described in Chapter 3. The second display was the Enhanced Graphics Adaptor (EGA) on the host machine capable of displaying limited resolution graphics. With two displays both were used in the user interaction, the imaging board for locator and valuator functions, the EGA for all the classical user interaction and feedback.

To an extent this involved a certain amount of shift in the attention of the user. However the advantages of using a dual screen display outweigh the disadvantages. The choice for a single display system would use the imaging board, which has 768 x 768 pixels with 24 bits/pixel colour. The menus and interaction could all be performed on this display, but would necessitate masking of the image with the interface screens. The system was therefore designed such that it was only necessary to use one display of the two at any one time. The selection of menu and function options, the input of function parameters and the display of numerical output is performed on the lower resolution EGA display. Only the core operation is performed and displayed on the image display.

7.3.1.1 The 'Locator'

The locator for both displays was a mouse driven cross hair cursor. For the EGA display the logical co-ordinates system and resolution of the screen corresponded to that of the mouse. The image display was however much higher resolution (768 x 768 pixels) and was oriented in a different manner. The problem is to relate the movement of the mouse on the pad to the movement of a cursor on the display. The design of the locator became the consideration of a number of factors. A library of mouse primitives was available which enabled the absolute position of the mouse, relative movement and the status of the button to be determined. The relative movement function returns the displacement of the mouse on the pad since the last call to the same function. Using this function a position can be determined by incrementing or decrementing a absolute position counter with the relative movement.

To provide feedback, the current position of the cursor is displayed on the EGA, in co-ordinates, and on the image display as a colour cursor. Updating the screens with new positions is a further concern with screen update flicker and response time the important factors. A routine was eventually derived with a fast response to mouse movement, positioning to single pixel accuracy and update of the screen only at significant visible intervals.

The locator routine makes use of the hardware pan, scroll and zoom features of the graphics display processor. Of the 768 x 768 pixel area only a 512 x 512 pixel window is used and the locator routine accounts for this by panning the hidden areas into view when the cursor is moved closed to the edge of the window. For all zooms greater than 1 the cursor is stationary at the centre of window and the image is moved. This maintains the point of concentration in the centre of the screen. Code fragment 1 in Appendix A is a listing of 'C' source for the locator routine.

7.3.1.2 The 'Pick'

The simulation of the 'pick' was achieved using a mouse driven cursor, each object is selected by moving the cursor to the object and pressing the left button on the mouse. This is also present in the system as selection of function options available.

This was performed using a routine similar to the locator with a nearest neighbour search or neighbourhood box applied to find the selected item. Typical application of the 'pick' in the system is in selecting a feature to manipulate.

7.3.1.3 The 'Valuator'

With the aim of reducing keyboard manipulation to a minimum, a valuator was developed to use for input of integer variables. The input of floating point variables and larger integer variables was achieved using the keyboard. The routine accepts the minimum and maximum values of a range and the default for the value to be input. Horizontal movement of the mouse to the left decreases the value, movement to the right increases the value. The sensitivity of the movement required for a unit change depends on the range of possible values.

This routine updates the EGA screen during each selection and is repeated until the left button is pressed, whereupon the value is assigned. Code fragment 2 in Appendix A is a listing of 'C' source for the valuator routine.

7.3.1.4 The 'Button'

The operation of the 'button' is to select between various options presented to the user. The button system developed uses the hierarchical tree structured menu system detailed in section 7.2.1.

Each pop-up menu presents a number of options, one of which is selected by moving a mouse driven arrow cursor. Each option is abbreviated on the menu but is described in a message line when the cursor points at the option. The option is chosen by pressing the left button of the mouse, whereupon the screen changes to the selected option output screen or a further menu.

The tree structure is never more than three deep and the number of options in each menu is kept to less than 10.

This particular form of command selection provided a fast, easy to use interface to the numerous system functions. Code fragment 3 in Appendix A is a listing of 'C' source for the menu routine.

7.3.1.5 The Keyboard

The main function of the keyboard in the system is to input character strings, corresponding to file names, or image labels, or to input floating point or large integer values. As the EGA is in a graphics rather than text mode this involved writing a keyboard handler routine. This routine accepts characters from the keyboard, checks for validity of the character and fills a character string buffer. The routine must also allow for deleting existing characters and editing of default strings. Code fragment 4 in Appendix A is a listing of 'C' source for the graphics oriented keyboard handler routine.

With certain functions the typing of file names at the keyboard has been replaced. This is achieved by interrogating the directory table of the current directory on the disc. From this a list of the relevant files is sorted and presented on the display. A filename is then selected by highlighting the required file with a mouse driven cursor. This has particular application in selecting files to read into image display components, and provides a rapid method of entering character strings from a list. The routine must also allow for paging of large directory lists. Code fragment 5 in Appendix A is a listing of 'C' source for the searching of a directory for files of a given extension and selection of a file from the directory list.

7.3.2 Error Management

In the Water Information System, the user interface is graphical rather than textual. The DOS critical error handler operates by outputs messages to a text display over writing graphics and causing an unsightly system failure. All possible errors must either be anticipated or trapped and a hardware error handler operating in graphics mode produced.

Errors are handled in two ways, the most common way is to expect possible errors and not allow the user to create the error. An example of this is in the keyboard handler for input of floating point values; the only valid input from the keyboard are the numbers 0-9, a minus (-) and a decimal point (.), any other input would cause the conversion from character array to floating point value to fail, and therefore the routine does not allow input of invalid characters. This error along with many others invoking a 'beep' and an explanatory message. This form of error handling is present throughout the system.

There are certain errors which cannot be handled in this manner, for example device not ready. These are errors at system level caused by, for example, a disk drive door not being closed, or a printer being off-line. By necessity the handler is device dependent.

These errors involve writing an error handler to replace the DOS error handler. DOS only allows a subset of the complete error handler to be replaced and this can be achieved using DOS Int 0x24h, and replacing the address of the critical error handler with the address of the replacement error handler.

This critical error handler deals with 13 errors including: 'drive not ready', 'write fault', 'read fault', and 'general failure'. The main error to be handled using this function is 'drive not ready'. This is done by loading the replacement handler, which returns an error code, and attempting an I/O operation on the drive. The error code is then analysed and consequently handled, with a graphics message. Code fragments 6 and 7 in Appendix A are listings of 'C' source for the error handling routines.

For a number of system errors, such as printer errors, service requests can be issued. The printer port status routine checks the printer for a number of errors using BIOS Interrupt 17h, Function 02h - request printer port status. The status of the printer port is returned in a register that can easily be decoded to give the nature of the error. Code fragment 8 in Appendix A is a listing of 'C' source for checking printer port status.

7.4 Data Formats

There are numerous commercially available mapping and information systems. These range from the Intergraph mini-computer systems and vector graphics workstations running MOSS or ARC/Info, to the micro-based systems running AutoCAD or similar draughting packages. There is a certain amount of overlap between digital mapping systems and conventional CAD packages, the importance being placed in the production of vector based spatial databases.

Any information system must therefore have the capability to handle data in numerous formats: the key to information systems is integration. Most systems provide this service as a utility to transfer prior to operation in to/out of an internal data structure.

Within the water information system, raster representations of thematic maps can be produced from the vector data by a process of boundary recall and seed fill. In this study , the raster thematic map is stored as a 1 byte/pixel sequential data file. The value of each pixel reflects the class, or theme, of the pixel, which can be extracted from a table of attributes.

The structure of both vector and raster databases is an important factor controlling speed, efficiency and storage requirements of the system.

7.4.1 The Vector Data Structure

For this study a data structure was derived to enable fast recall, sorting and editing of features. This structure is used for storage of vector data, such as linear features and boundaries, within the water information system.

The data structure is essentially an indexed sequential file. The index is provided by a header structure containing a number of attributes of the feature including the feature type, feature code, feature number, number of points making the total feature, representative colour, and a deletion marker.

There are 5 classes of feature type:

1. Line
2. Area
3. Symbol
4. Contour
5. Spot height

The contour and spot height are special forms of line and symbol features.

The feature code is a four figure classification of the feature and relates to a user defined hierarchical feature code tree. The feature number enables a fast skip through the file when searching for feature data.

Each feature is organised as a header structure followed by a sequential list of x,y co-ordinates. The feature is redrawn by drawing straight lines between the points, thus each line segment is defined by the previous and current point in the vector list. Areas are stored as a boundary list and seed fill point.

An example of the vector data structure is given in Table 7.7.

Vector Data			Description
0 9 0	0.0	1 1100 1	Attributes - COL NP FT ZP FN FC FL
245320	125873		
245330	125910		
245335	125995		
245342	126052		
245355	126132		Feature vertex list (OS Grid)
245365	126152		
245385	126187		
245407	126232		Where :
245420	126257		
12 9 0	0.0	2 1140 1	COL - Colour of feature
245277	126445		NP - Number of points in feature
245275	126450		FT - Feature Type
245247	126455		ZP - Height of feature (contour data)
245245	126455		FN - Feature number (Index)
245247	126510		FC - Feature classification code
245267	126520		FL - Deletion Flag (0 Off, 1 On)
245290	126517		
245292	126517		
245312	126445		
1 2 0	0.0	3 1150 1	
245962	126370		
245980	126387		
10 6 0	0.0	4 1860 1	
245875	126377		
245882	126362		
245905	126375		
245900	126382		
245885	126387		
245875	126377		
13 1 2	0.0	5 1710 1	
245078	126178		

Table 7.7. The Vector Data Structure.

7.4.2 The Raster Data Structure

There are a number of ways a raster thematic map can be stored. The map is distinct from an image or surface map in that a number of areas of a single brightness values make up the total database. The map can therefore be stored as a set of lists of boundary co-ordinates. This probably represents the greatest data compression of a raster map. There also exists a number of ways the raster map can be encoded to achieve compression, including run-length and quadtree encoding of the data. Data compression techniques are required when storage space is limited. With the advent of optical storage devices and the falling cost of conventional storage in large volumes, this is no longer seen as a restriction. Perhaps of more importance is the speed of access to the database. Decompression of data, depending on the compression technique, can be slow.

7.4.2.1 Run Length Encoding

Run length encoding of data is perhaps the most simple form of data compression that does not reduce the spatial resolution of the data. As the name suggests the data is stored as run lengths of pixels. The compression is 1 dimensional and simply reflects the change in value of a pixel. The data stored is the number of pixels in a run of a single value and the pixel value. For example the following sequence:

```
GGGGGGGRRRRRRUUUUUUUSUUGGGGEEEEEOOOOOOOLLLLLGGLLGGMMMMEEEEEEERRR
            < run >
```

- is stored as:

```
7G 6R 7U 1S 2U 4G 5E 7O 5L 2G 2L 2G 4M 7E 3R
```

Each run is stored as two values. For 8 bit image map data each item of data is a byte giving a maximum length of 256 pixels of a value between 0 and 255. Each run is 2 bytes therefore the above example would compress from 66 bytes to 30 bytes. If any particular run is longer than 256 pixels the run is divided into 256 pixels lengths. Thus an area 512x512 pixels of one value is represented by 2048 bytes (512 x (2 + 2)), as opposed to 262144 bytes uncompressed.

Image data does not compress unless there are large contiguous blocks of brightness. If every pixel along a line of pixels is distinct from its neighbours, then each line would be represented by 1024 bytes (512 x 2), and the 512x512 image stored in 524288 bytes.

A simple routine to encode and decode data is presented in Code Fragment 9 in Appendix A. Table 7.9 gives details of timings and storage requirements for a sample data set using this routine.

7.4.2.2 Quadtree Encoding

Quadtree encoding of map data is a 2 dimensional method of data compression, and is achieved by representing an irregular area by a series of rectangles of varying sizes. Traditionally a quadtree encoding of data is described as dividing an image into quarters. Each quarter is then examined and is further divided if there is variation within each quarter. This process continues until there can be no further subdivision, or the lowest level of division is reached - pixels. The simplest form of quadtree is a binary quadtree, that represents only a single class division, and requires only position and block size to be encoded. For multiple class representation a class value must also be encoded. Figure 7.3 demonstrates the process of encoding a simple class map using quadtrees.

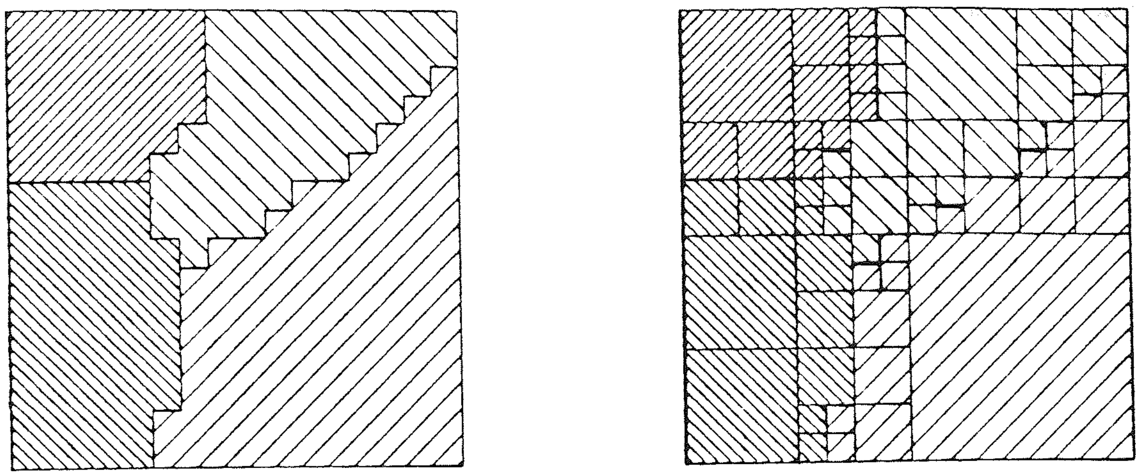


Figure 7.3. Quadtree Encoding of Spatial Data.

Once a contiguous block of data has been found its corner co-ordinates, size and value are all that are required to represent the data. Due to the recursive nature of the problem, it is conceptually simpler to program quadtree encoding from the finest resolution to the coarsest. This involves first examining the image in blocks of 2x2 pixels if there is variation within the block then each value is stored along with its position and the block masked to prevent inclusion within larger blocks. Thus all the variations of a single pixel are found first - a rough boundary edge map, and the remaining image is now composed of contiguous blocks of 2x2 pixels. The process then repeats for each level of block size.

Quadtree encoding is much more dependent on the shape of the areas and their alignment to the dividing boundaries. For a 512x512 image with a single value the storage requirements are simply the bottom-left pixel co-ordinates, the value and the size, i.e. 0,0,128,512. Image data does not compress well and at the worst case may involve storage of 4 values for each pixel.

Code Fragment 10 in Appendix A is a routine for quadtree encoding and decoding of spatial data. For speed and flexibility of retrieval, each contiguous block of image data is represented by a 32 bit word, containing a 24 bit x-y block offset and an 8-bit class value. The words are sorted in order of block size and the number of blocks of each size stored in a 12 word header. In this way each set of blocks can be retrieved independently, or the image sub-sampled by retrieval of all block sizes larger than a given size.

Table 7.9 compares the various techniques of data compression, in relation to timings and storage requirements. Quadtree techniques can be seen to have no real advantage over other methods for speed or storage. However because the quadtrees encode spatial data rather than linear data, re-scaling and sub-sampling of data is more efficient. Sub-sampling of data (on a bit boundary) being a simple case of reducing block size and losing the lower resolutions, lower branches, of the quadtree.

7.4.2.3 Class Coding

The brightness value of each pixel relates to the class it depicts. For 8 bit/pixel data each raster map can 256 possible classes. The attributes of each class are stored in a separate indexed sequential file, known as a map class table.

The raster map can be one of two types of map, either a class map, such as geology or soils, or a surface map, such as a digital elevation model or rainfall intensity. Surface maps can also be processed as class maps by representing ranges of brightness as a class. For a surface any number of class tables can therefore be used.

Map Class Table	Description
Land Use	Title of Map Class Table and Key
5	Number of Classes in table
0 32 1 0 0 255	Class Attributes - CN CV CR R G B
Water	Key Descriptor
Water Bodies, Lakes	Line 1 of feature descriptions
Streams, Rivers, Estuaries	2
Coastal	3
?	4
?	5
1 64 1 255 165 165	
Residential	Where :
Residential built up areas	
?	CN - Class number
?	CV - Class grey scale value (1-254)
?	CR - Class range CV to CV+CR-1
?	R - Red component of class colour
2 96 1 165 255 255	G - Green component
Industrial	B - Blue component
Industrial built up areas	
?	
?	
?	
?	
3 128 1 165 255 165	
Recreational	
Recreational areas	
playing fields, common land	
?	
?	
?	
4 160 1 255 255 165	
Agricultural	
Agricultural Lands	
grazing pasture	
fallow and cropped fields	
?	
?	

Table 7.8. The Map Class Table.

Each entry in the table comprises the class pixel value, or range of values, its representative colour, its key descriptor and a descriptive list of attributes of each class. These descriptors and attributes are in a format enabling conversion to, and analysis in, a relational database system.

An example of the Map Class Table format is presented in Table 7.8.

7.4.2.4 Data Retrieval

The relationship of the data to class values and location is provided by a header file. This file provides the size of the original map file, the compression technique used, and points to the table file defining the classes of the map file. Figure 7.4 shows the relationship of the map class files within the system of recall.

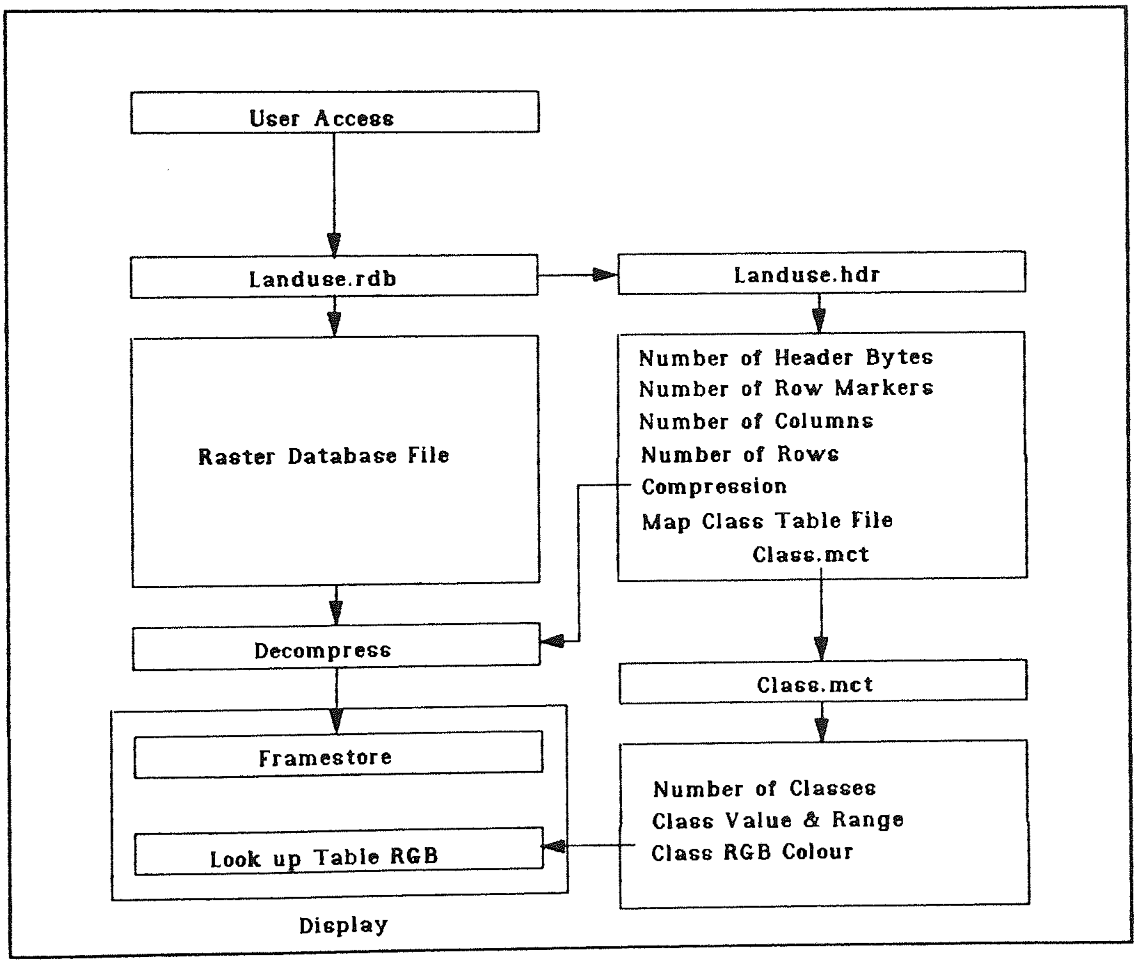


Figure 7.4. The Raster Data Structure.

Recall of data depends on the method of compression used, for all processing data must first be decoded and stored on the framestore. This allows application of spatial analysis on the framestores due to the immediately apparent spatial relationship of pixels. Using the algorithms described in Sections 7.4.2.1 and 7.4.2.2, Table 7.9 shows the storage requirements (KB) and time taken (s) to encode and decode samples taken from the Magat data set (see Chapter 8).

Data Set	Size (pixels)	Compression Technique								
		None			Run Length			Quadtree		
		Save	Read	KB	Enc.	Dec.	KB	Enc.	Dec.	KB
Geology	512x512	3.0	2.1	262	3.4	3.0	15	83.6	11.8	90
L.M.U	512x512	3.2	2.0	262	3.6	2.7	24	87.0	16.8	129
Soils	512x512	3.3	2.2	262	3.4	3.0	16	84.5	12.1	92
Land Use	512x512	3.1	2.1	262	3.5	3.0	20	86.1	14.6	111
Rivers	512x512	3.2	2.1	262	3.8	3.1	38	92.1	25.2	194

Table 7.9. Data Compression and Retrieval, Time (s) and Storage (KB).

From these results it would appear that for both speed and compression run length compression techniques should be used. However there are certain situations where quadtree compression techniques have distinct advantages. For example, if the data is to be rescaled, by a factor of 2, the speed of retrieval increases for uncompressed and run length encoded data, and decreases for quadtree encoding.

8 SYSTEM APPLICATIONS

There are numerous potential applications for integrated information systems and remote sensing in the fields of hydrology and water resources. A few sample applications were used as an aid during the user requirement specification stage of the design and in the overall development of the water information system.

The sample applications cover a wide range of remote sensing and water resources management activities; from large scale airborne video to satellite remote sensing; from drainage network design to drainage basin pollution monitoring and erosion risk models.

8.1 Creation of Hydraulic Model Input Data

The first application involves the generation of data for hydraulic models, in particular production of Digital Elevation Models for tidal and estuarine water quality models, and classification of contributing areas for input to the WASSP drainage design program.

Input to both types of models can be the most time consuming and costly part of the process. Conventional methods involve the digitisation of bathymetric charts and topographic maps using bit pad digitisers and file based interpolation. The advantage of using integrated spatial information systems is the increased efficiency of these operations and the integration of these data with data from other sources. A by-product of the production processes is the advanced display of model results.

8.1.1 Tidal and Estuarine Model Input Production

Conventional methods of producing input for grid-based hydraulic water quality models uses an operator who manually follows contours using a cursor on a bit pad digitiser, driven by a CAD or Digital Mapping system. This results in a file of x-y-z co-ordinates which are then interpolated onto the required grid.

Models are generally produced from several sources: survey collector charts, admiralty charts and topographic maps. Each could be of a different scale, have different datums and units, and be produced in a variety of map projections. The digitisation system must be able to produce a data set on a common geographic base, with a single unit of measure related to a single datum. In addition confusion must be resolved in areas of overlap between charts.

This can be provided by the basic functionality of an integrated spatial information system, similar to that described in this thesis.

The process involves video digitisation for input of cartographic data and interpolation of these data to a grid base. The various procedures which comprise the overall production process are represented in Figure 8.1.

8.1.1.1 Chart Digitisation

The first stage in the process involves the preparation and video digitisation of the chart. Preparation of the chart consists of marking grid control on the chart and in some cases manually drawing contours on the chart based on survey depth soundings. Preparation may also involve photo-reducing the chart to more manageable proportions.

Grid control data is subsequently used to determine, and correct for, geometric distortion introduced in the preparation and digitisation process. The result is a chart geometrically referenced to the model grid co-ordinate system. Figure 8.2 shows the processes involved in chart digitisation.

8.1.1.2 Contour Digitisation

The options for contour digitisation from the charts has already been discussed in Chapter 4. The simplest method is similar to conventional methods using bit-pad digitisers; the contour is traced with a cursor driven by the digitisation software.

The software developed for this process enables contours to be deleted vertex by vertex, or completely. The completed contour can subsequently be moved, copied or deleted, vertices added, re-defined or deleted. This editing process is performed on-screen, where the topological relationship of contour data from adjacent charts can be examined. Figure 8.3 and 8.4 show the processes involved in digitisation and editing of contour features.

8.1.1.3 Chart Merge

For each chart a list of attributes are recorded. Some are used in the digitisation process to reduce the data to a common base. Additional information such as date surveyed and the spatial extent of data input from each chart are used in the merge process.

Prior to recall of data for interpolation the priority of data in overlapping areas must be considered. This can either be under control of the operator or can be defined by some automatic method, such as chronology.

Merging of data is achieved by producing a priority map. The spatial extent of data from each chart is defined by a boundary. A filled polygon defined by the boundary from each chart is then recalled in chronological order, the data from the most recent chart taking highest priority. This priority map forms the basis for an interactive session, whereby each area of overlap is examined and its priority redefined by the operator. Figure 8.5 shows the merge process.

8.1.1.4 Interpolation

The interpolation of contour data onto a grid base has already been described in Chapter 5. For the purpose of depth modelling, interpolation methods such as least squares and multiquadric, are generally too slow and over precise. Simpler algorithms such as bi-linear interpolation of Triangulated Irregular Networks (TIN) provide fast interpolations of sufficient accuracy, but require a specific data format. In this particular case, the generation of the TIN involves far more computation then the interpolation, and there may well be no time saving.

Invariably 8-bit data is insufficient to model the variation of depth. For this reason the 32-bit memory of the framestore is divided into two 16-bit components, input data and output surface. In this way floating point values can be represented as rescaled unsigned integers. As an example a range of 65 m can be modelled with a 1 mm height step.

8.1.1.5 Conclusion

In terms of overall time taken to produce input for hydraulic models, the use of the water information system may not provide a significant saving until automatic contour extraction processes are refined. The main advantages of its application lie in the nature of the data structure, which can then be used as a common base for additional survey and modelling work.

The range of possibilities for display of model output are far in advance of conventional methods. Any source of information, point source or spatial, vector or raster based, can be integrated with model output. For example, output from water quality models can be coloured and displayed merged with photographic and digital remotely sensed imagery. Isometric and perspective views of the merged data sets can be generated and multi-temporal model data can be animated at specified time steps in any merged form.

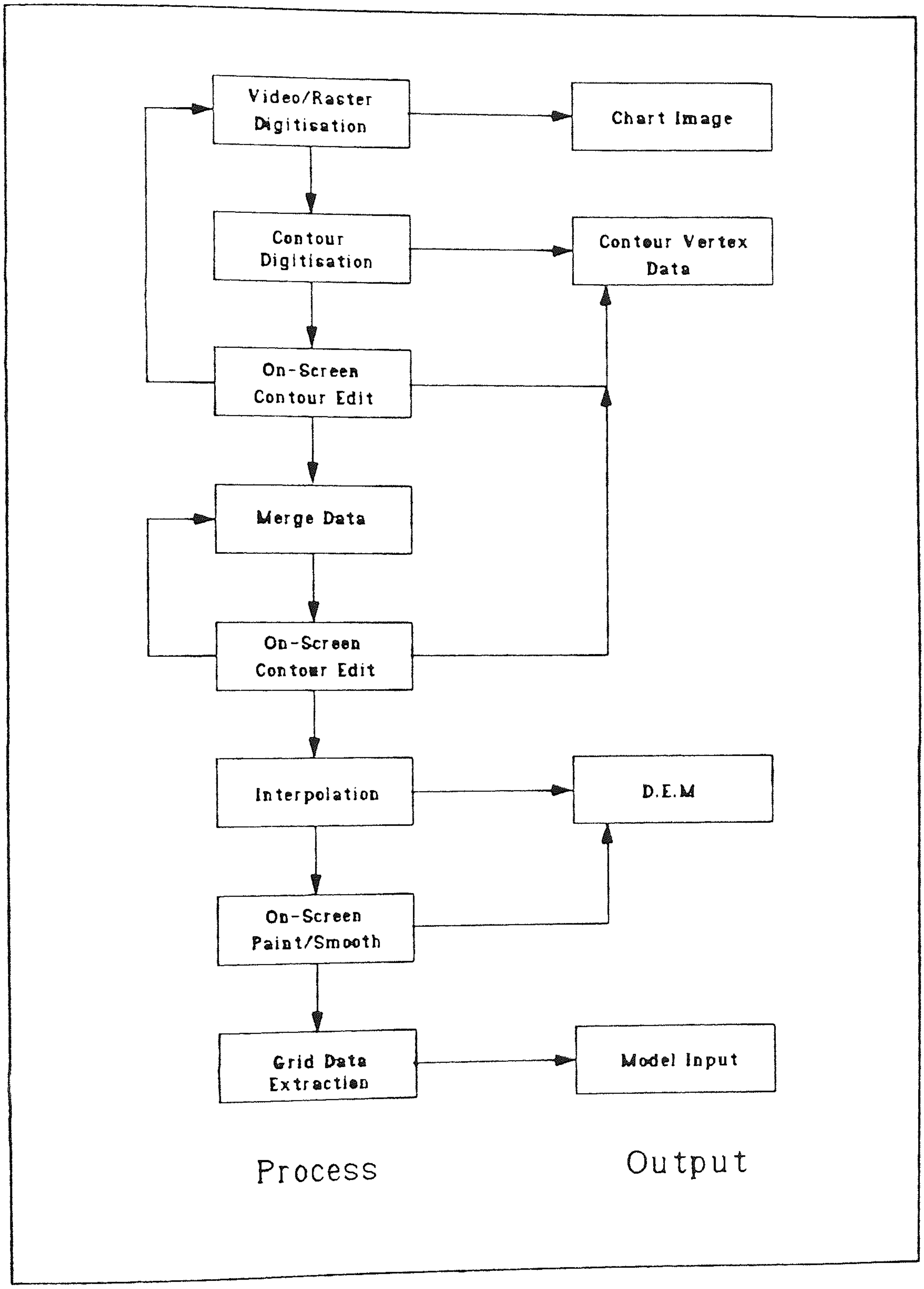


Figure 8.1. Generation of Hydraulic Model Input Data.

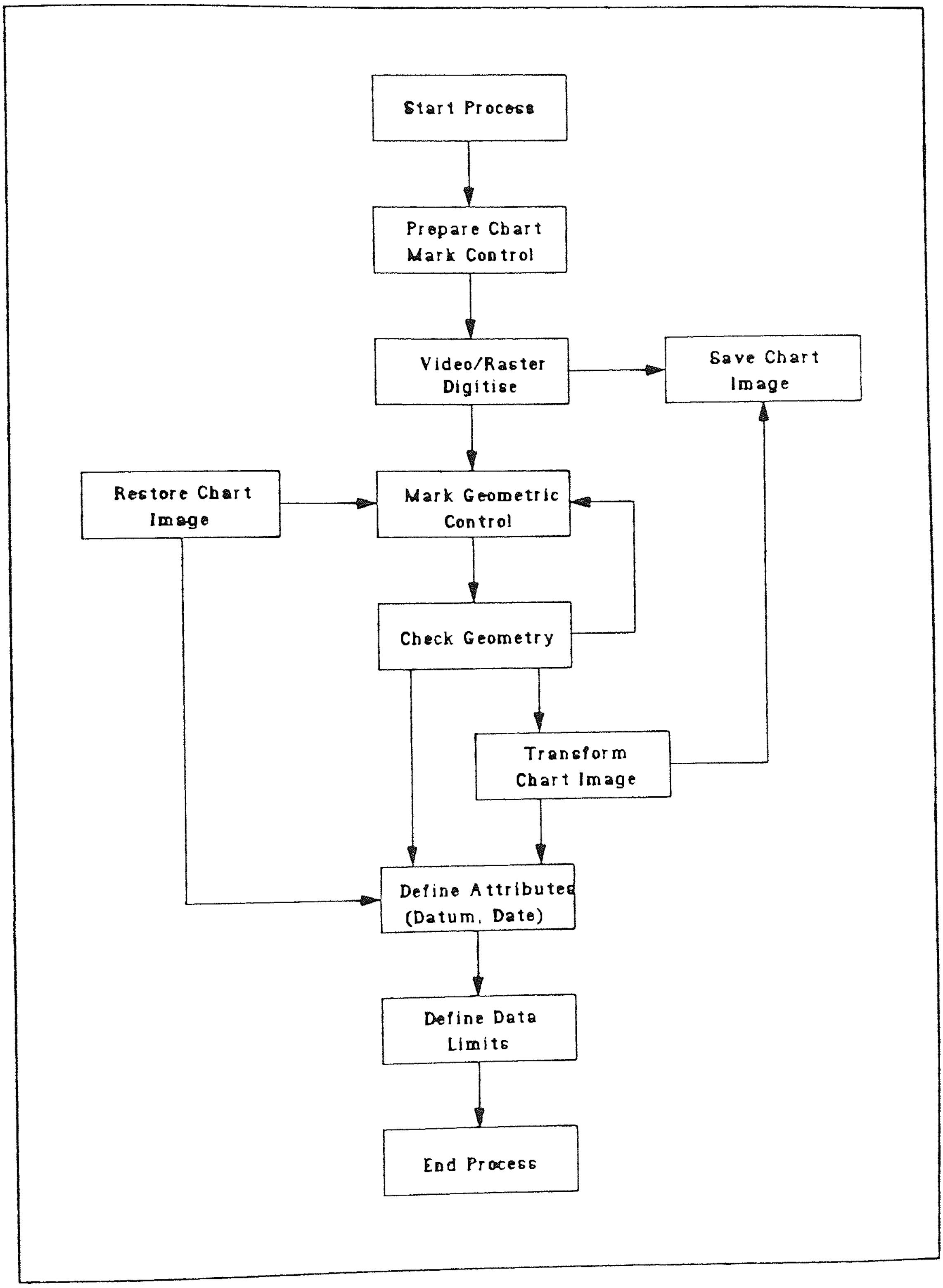


Figure 8.2. Chart Digitisation.

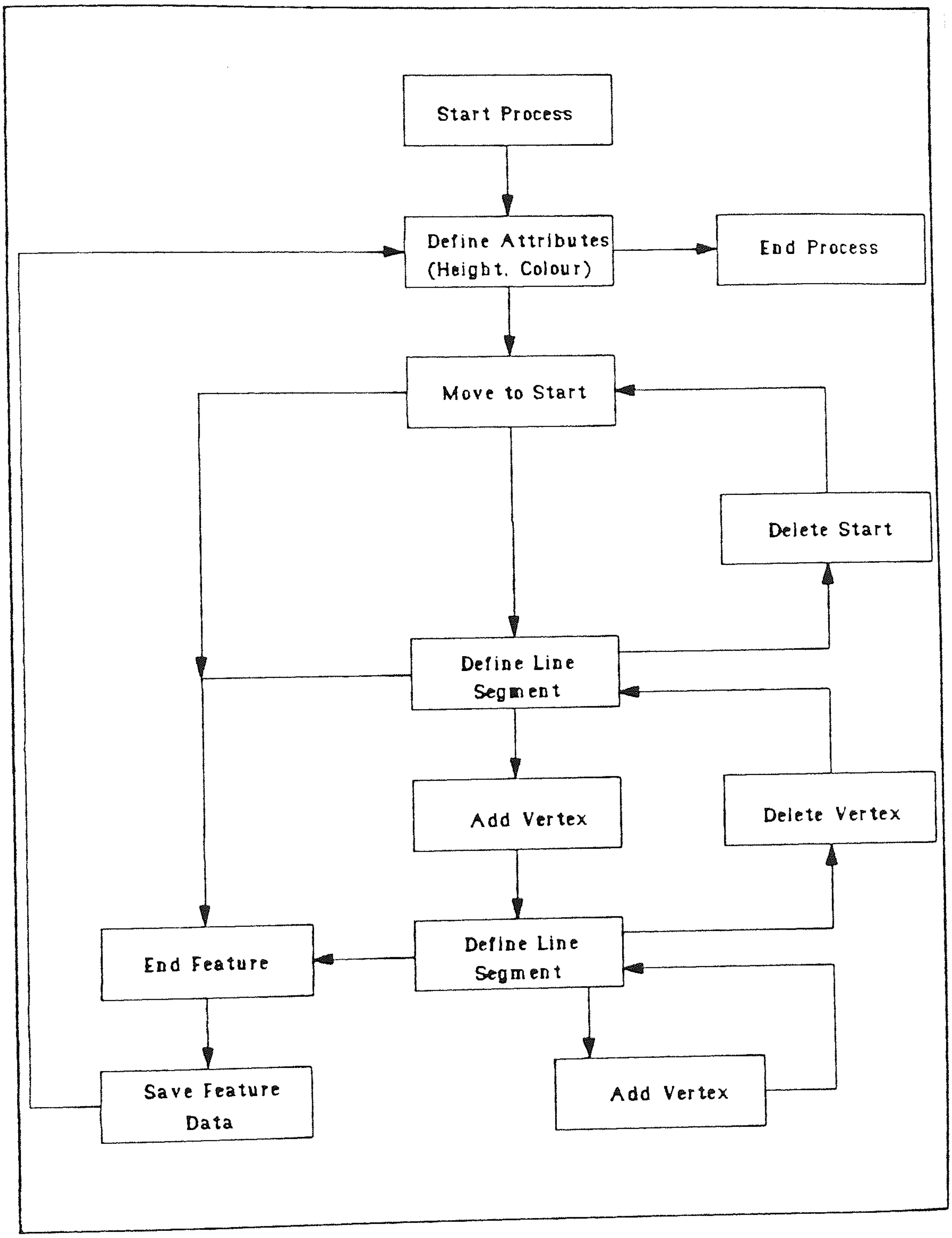


Figure 8.3. The Contour Digitisation Process.

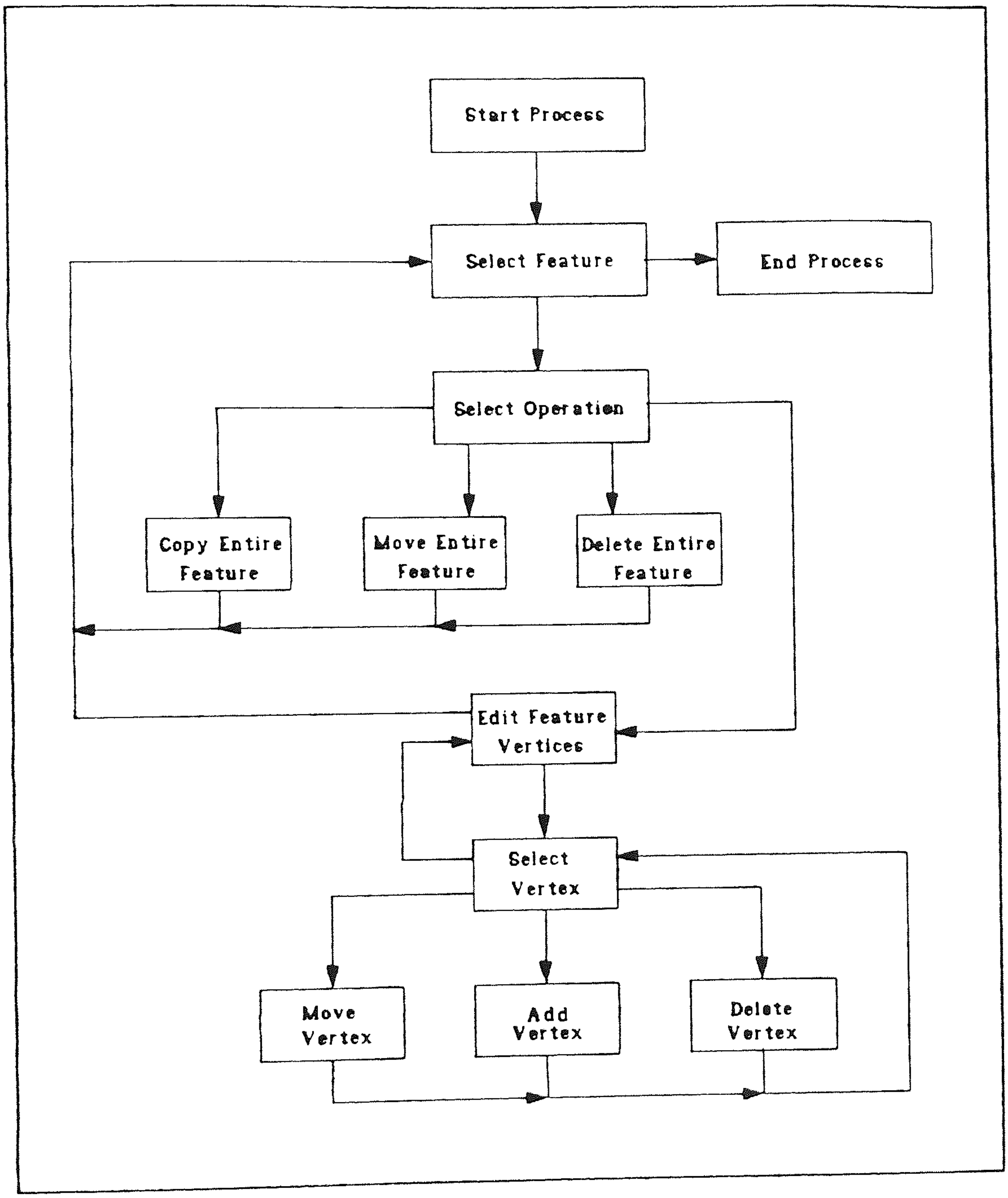


Figure 8.4. The On-Screen Contour Edit Process.

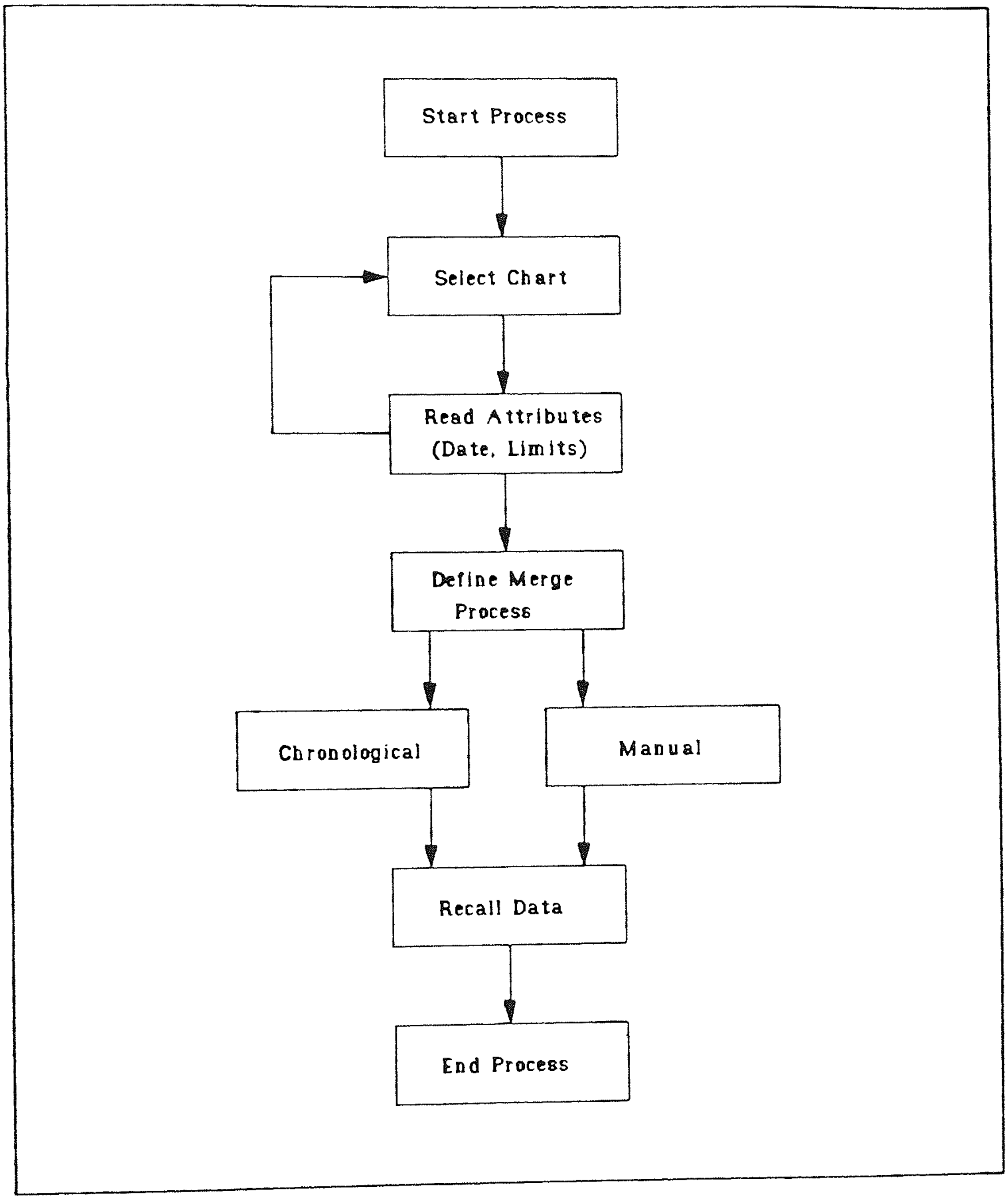


Figure 8.5. The Chart Merge Process.

8.1.2 WASSP Input Data Production

As hydraulic models develop in complexity, the requirement for data increases, this is the case for distributed hydrological models such as that described in Section 8.4, and also drainage network design and analysis systems such as WASSP (Wallingford Storm Sewer Program).

The WASSP system designs pipe networks to cope with a specified intensity of storm. The main problem facing potential users of WASSP is the generation of input data to run the model. The model itself executes in minutes, it may however take weeks to generate sufficient input data.

Given a proposed drainage network the creation of input data involves a number of processes. Figure 8.6 details the processes involved.

8.1.2.1 Drainage Network Layout

The first stage in the process is to overlay the drainage network on the topography. Conventionally this involves plotting the network on topographic maps. In the water information system the topographic maps are video or scan digitised, as described in Chapter 4 and in Section 8.1.1.2. The drainage network is then represented as vector lists of pipe data, giving co-ordinates and attributes of each pipe length and node, and recalled onto the geographically referenced raster digitised map.

8.1.2.2 Watershed Partitioning

The next stage in the process is to divide the topography into watersheds contributing to each pipe length and node. This can be achieved by a number of methods. The simplest method, and probably the quickest, is to manually define the watershed boundaries. An alternative to this is the automatic partitioning of the topography as described in Chapter 5. This would require the production or acquisition of a Digital Elevation Model and may therefore prove too time-consuming and expensive.

8.1.2.3 Determination of Contributing Areas

The final stage in generation of input data is the areal measurement and classification of impermeable areas within each watershed. At this stage the conventional and proposed methods differ considerably.

Conventionally this is achieved using a bit-pad to digitise the boundaries and hence calculate the areas of each roof top, road and other impermeable areas. This process can be extremely laborious and time consuming.

The proposed method relies on image processing and graphics techniques to reduce the necessity for digitisation of permeable/impermeable areas. This is achieved using the video digitised topographic map and derived watershed boundaries. The process is detailed in Figure 8.7.

A video digitised map of sufficient quality is prepared such that variable lighting effects across the watershed are not too severe. The watershed boundary is recalled and the areas outside the watershed are graphically removed. The line information within the watershed is then binarised and re-scaled, such that lines are represented by a pixel brightness of 0 on a background of pixel brightness 255. The threshold for the binarisation can be defined interactively or by automatic methods, with either, the time taken to achieve this process is negligible.

Once binarised the contributing impermeable areas are defined by a seed fill process, where the operator points to an area with a cursor, and defines it as a class of impermeable area. Once all impermeable areas are defined the lines are removed from the image using a convolution operation that replaces pixels of the line colour with the median of the line pixel's neighbours. Statistically this operation maintains the relative proportions of each class.

The number of pixels in each class is then counted, permeable areas defined by exclusion from impermeable classes, and the area and percentage of each class determined. Figure 8.8 represents a sample data set derived from an OS 1:10560 topographic map using this method.

8.1.2.4 Conclusion

The overall time saving in production of input data by this method is perhaps its major advantage. Further possibilities exist in extraction of data from the raster digitised map produced.

A number of existing systems such as the BBC's Domesday Project and British Telecom's Cadastral Mapping system use video disks to store images of topographic maps at various scales. These and other similar systems use video displays, enabling a simple interface to spatial information systems to be achieved using the methods of video digitisation described in Chapter 4.

The use of large scale airborne video also has applications in this field. The video imagery can be corrected to the map database, and automatically classified to produce maps of permeable/impermeable areas.

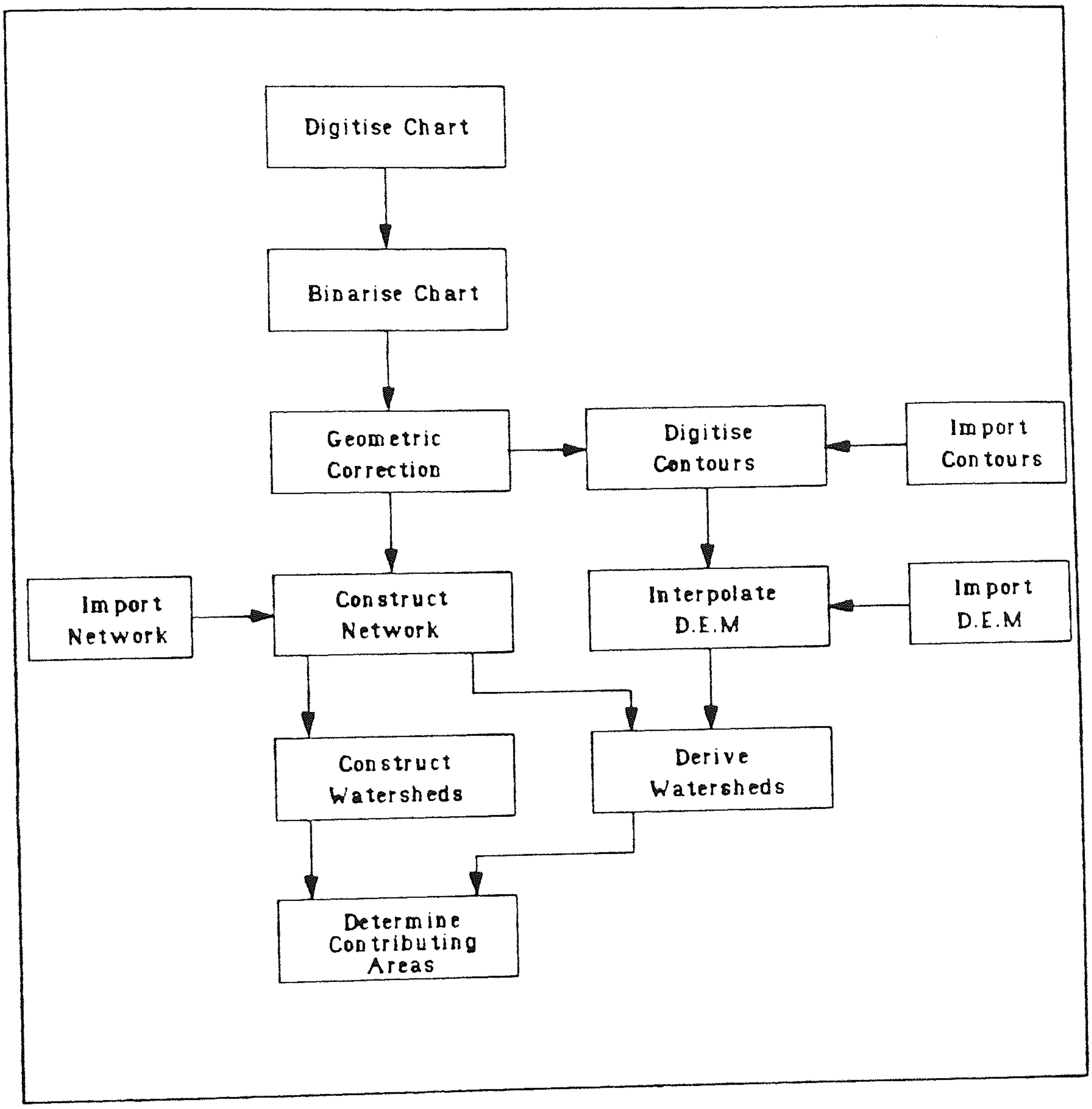


Figure 8.6. Generation of WASSP Input Data.

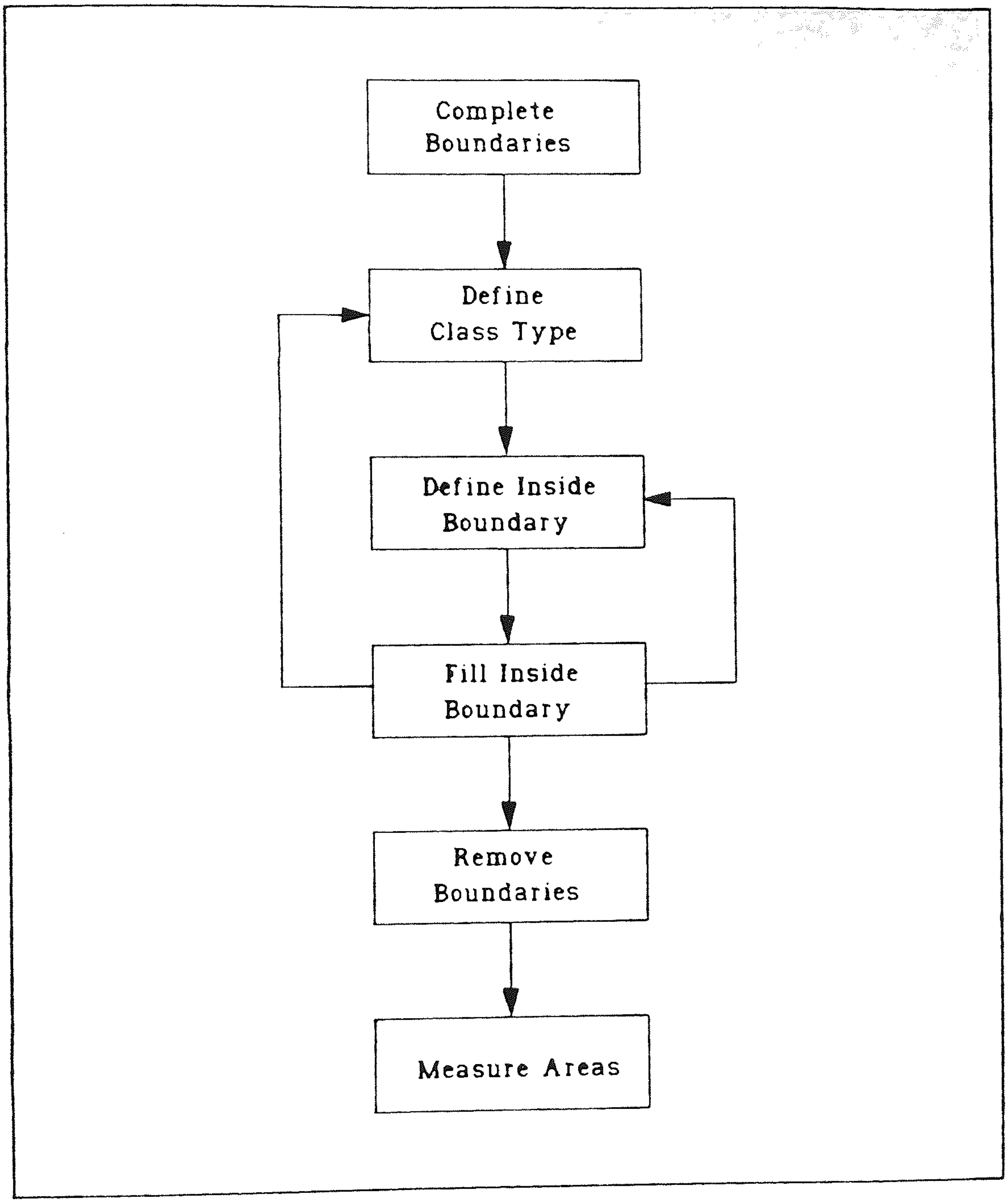


Figure 8.7. Determination of Contributing Areas.

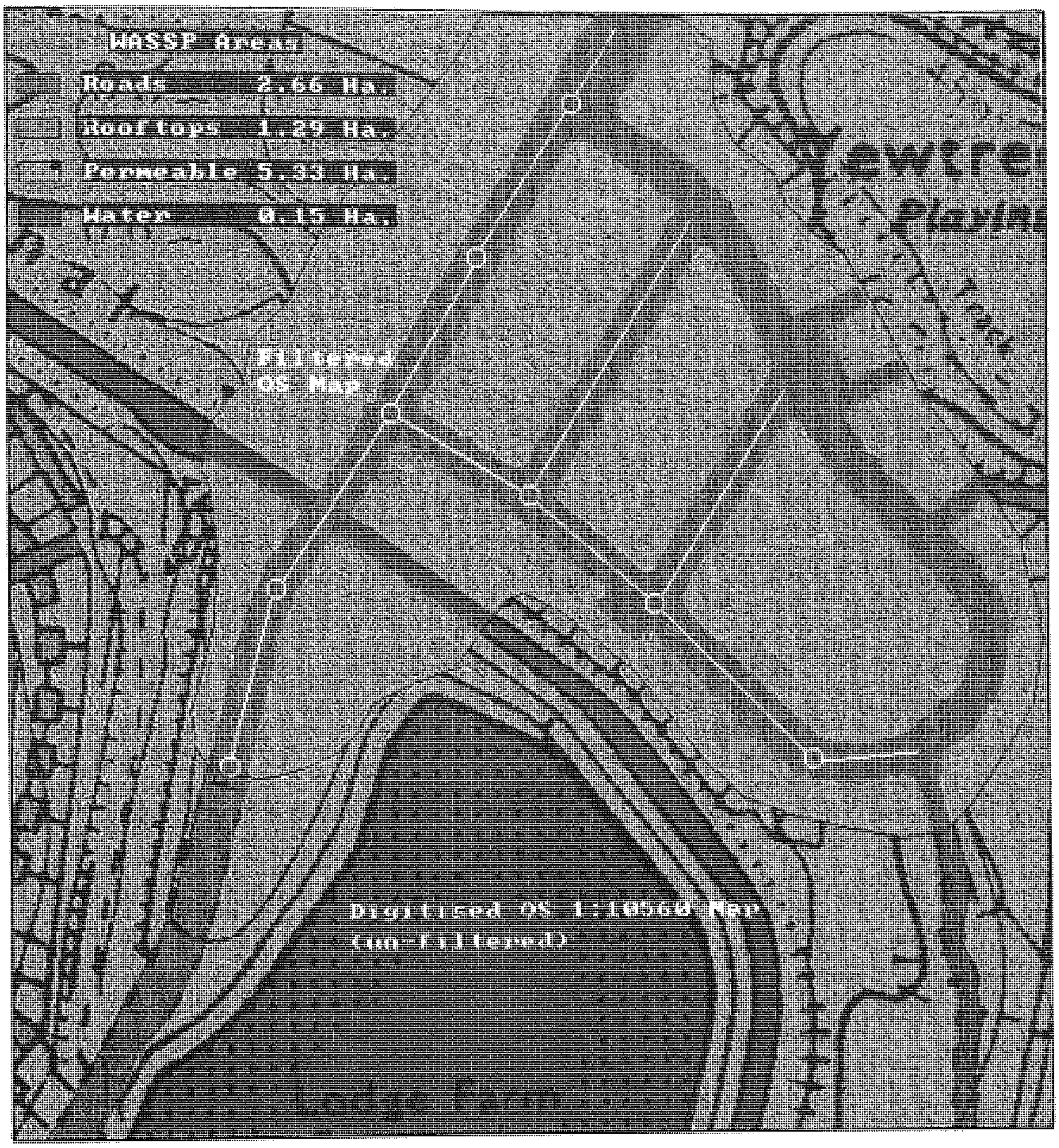


Figure 8.8. WASSP Input Sample Data Set.

8.2 Taw-Torridge Water Quality Survey

This work was carried out as part of the research into the feasibility of using airborne video data acquisition systems. It was undertaken with available equipment corresponding to the first system described in Chapter 6.

8.2.1 Introduction

To assess the feasibility of using airborne video as a remote sensing system, in particular for water quality mapping, a survey was undertaken to coincide with a ground based water quality survey being carried out by Wimpol Laboratories.

The water quality survey, undertaken by Wimpol, was commissioned by the South West Water Authority to comply with the Department of Environment's requirements, prior to consent being given to discharge screened sewage into the estuary. The DoE required that South West Water produced a comprehensive plan for water quality supported by adequate monitoring of both water quality and effluent discharges over a period of several years.

An initial intensive survey including bed grabs and water quality samples, temperature distribution and flow characteristics was carried out in the early stages of the plan. Over the last few years Wimpol have continued to carry out surveys in each season over both neap and spring tides at high water. It was these seasonal surveys that we intended to monitor using the airborne video.

Typical measurements being taken from the water samples were dissolved oxygen, turbidity, acidity, fertilizer pollution, ammonia from sewage, metals, organic compounds and biochemical oxygen demand (BOD).

Wimpol had expressed an interest in the airborne video survey and were prepared to release information to provide calibration for water quality models derived from the airborne video. The most likely measure of water quality that could feasibly be determined from the airborne survey is turbidity, or its surrogate - water colour. Colour is also likely to be affected by water depth, consequently any variation observed could equally be related to depth or turbidity.

If quantitative assessment of turbidity proves unlikely, then some form of qualitative distribution of turbidity could be determined.

8.2.2 Remote Sensing of Water Quality

Initial documented studies into remote sensing of water quality began in the late 1960s. These early studies relied on visual or densitometric analysis of photographic imagery. Scherz (1969) suggested that remote sensing techniques could be applied to the study of pollutants in river and estuarine water, and attempted to derive the optical characteristics of water containing various industrial wastes. The spectral resolution of the signatures was limited by the use of photographic media.

Later work by Klooster (1974), Lillesand (1975), Bhargava (1983), attempted to correlate measured turbidity and chloropyll-a with densitometric analysis of photography. Turbidity is dependent on scattered light from suspended particles, and is therefore related to the wavelength of light and the size, shape, concentration and reflectance properties of the particles. Turbidity is therefore constant for a given particle and lighting condition.

The field and laboratory measurement of the spectral response of natural waters containing sediment continued with spectroradiometers, Blanchard (1973), Bartolucci (1977), Whitlock (1978), Witte (1981) and McKim (1984). The effect of bottom depth, water surface roughness and sediment concentrations on turbidity were analysed and relationships derived. From Figure 8.9 it can be seen that the optimum wavelength for measurement of suspended sediment, with the greatest penetration is about 575 nm, visible green.

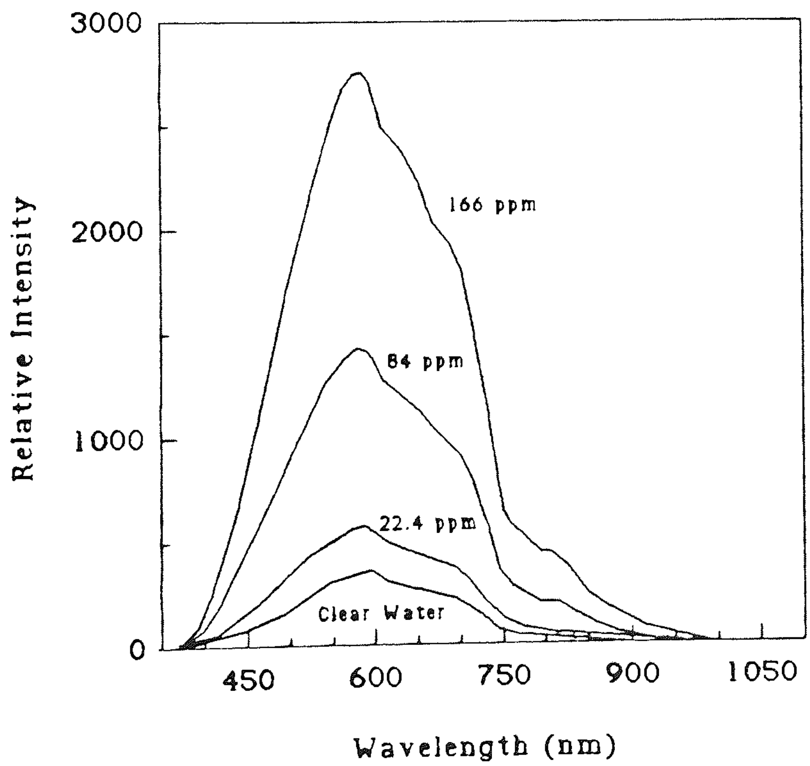


Figure 8.9. Spectra of Water Containing Sediment, McKim (1984).

This work became more significant in 1972 with the advent of digital remote sensing systems, commencing with the Earth Resource Technology Satellite - ERTS 1, later Landsat 1. The synoptic viewpoint afforded by satellite systems and the advantages of computer enhancement and processing of the measured radiance meant that water quality could be studied over much larger areas.

Klemas (1973) documented an early application of ERTS 1 data in observation of suspended sediment dynamics in Delaware Bay. This application has been extended to global proportions with the use of Meteorological satellites Tanchotikul (1987).

The quantitative measurement of water quality parameters, such as turbidity and chlorophyll-a concentration, depends on adequate correction of various contributing factors such as atmospheric attenuation and surface roughness. This is particularly important if multi-temporal survey is required.

Given one scene and calibration data, models can be derived relating measured radiances in spectral channels using regression or least squares techniques. This enables the spatial distribution of water quality parameters to be determined. This is well documented from the advent of satellite digital imagery to the present day, Clark (1974), Ritchie (1976) and Khorram (1981) using Landsat MSS data; Cheshire (1985) and Lathrop (1986) using Landsat TM; Johnson (1978) and Collins (1984) using airborne multispectral scanner data.

To increase the accuracy of these predictions the contribution of atmosphere scattering to measured radiances over water has to either modelled or accounted for. One documented method relies on the use of colour space measurement of brightness. If the effect of variable light conditions and atmospheric constituents can be measured, they can be removed.

The theory and application of chromaticity techniques have been documented Munday and Alföldi (1975,1978,1979), Lindell (1981,1985), Munday (1983) and Bukata (1983). Chromaticity methods can be applied to any data set, and therefore its application in correction of multi-temporal airborne video imagery is possible. For this reason a brief review of the theory is presented.

Over water the total radiance detected is given by:

$$L = L_p + T\left(L_v + L_s + L_g\right)$$

Where:

- L_p - Atmospheric path radiance
- L_v - Water volume radiance
- L_s - Surface diffuse reflection radiance
- L_g - Sunglint radiance
- T - Transmission of the atmosphere

Total radiance is related to the digital number recorded for each pixel and is determined from measured components or available sensor calibration statistics. The chromaticity co-ordinates X and Y can be derived from the radiances of three wavebands of data. For Landsat MSS, bands 4,5 and 6, for airborne video the Blue, Green and Red wavebands are used.

$$X = \frac{L_4}{L_4 + L_5 + L_6} \qquad Y = \frac{L_5}{L_4 + L_5 + L_6}$$

Where:

L_4 - Radiance of Landsat MSS Band 4. Similarly Bands 5 and 6.

Data sensed in infrared wavelengths, such as Landsat MSS 7, are not used. This is because the volume radiance, the required measure of turbidity, is insignificant at these wavelengths, water bodies absorbing virtually all infrared radiation.

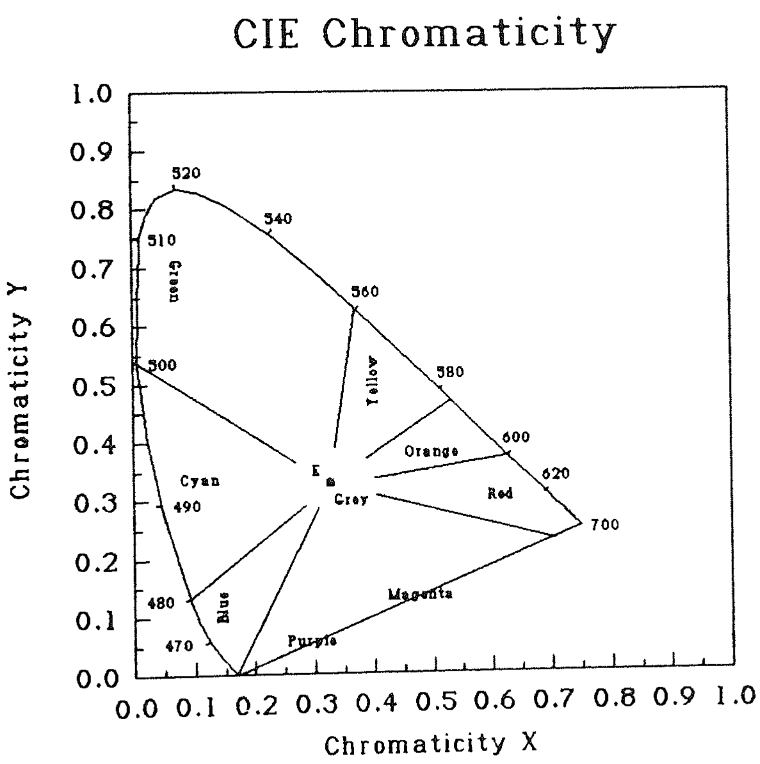


Figure 8.10. The CIE Chromaticity Diagram.

These co-ordinates when plotted on the CIE Chromaticity diagram define precisely the colour of the measured radiance. Figure 8.10 shows the standard CIE chromaticity diagram. The point marked E on Figure 8.10

is the 'white' point, co-ordinates 0.33 - 0.33, the point of equal intensity of the three components. Angular position about E is representative of the dominant wavelength or hue of the co-ordinate; radial distance from E the saturation or spectral purity of the hue.

The effect that variable atmospheric conditions and variable sediment concentrations has on position of the chromaticity co-ordinates has been studied and is presented in Figure 8.11. It can be seen that variation in suspended sediment characteristics and concentration define a locus forming a an arc around E. Variation in atmospheric conditions results in a radial shift to or from E. Thus atmospheric haze is determined to be 'grey' an increase in haze results in movement of the locus towards E, Munday (1983).

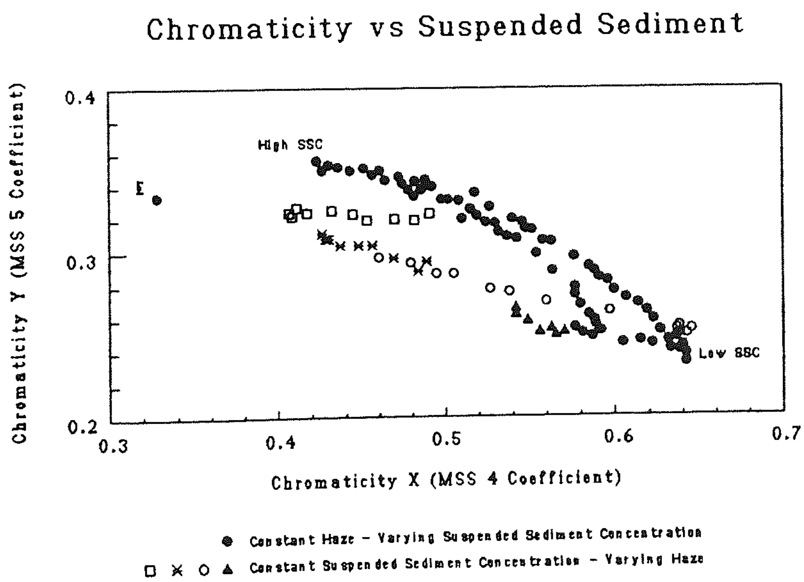


Figure 8.11. Variation of Chromaticity with Atmosphere and Sediment Concentration, Munday (1983).

Using this relationship a standard water locus can be derived from calibration ground data for a constant atmospheric condition and sediment particle. A measure of turbidity can then be derived for any atmospheric condition by shifting the co-ordinate of each pixel radially to or from E until the locus is intercepted.

The main assumption in use of chromaticity for measurement of turbidity is that variation in atmospheric condition and lighting causes a monotonic variation in chromaticity co-ordinate. Variation in turbidity is related to a variation in measured hue or dominant wavelength of the radiation. Consequently conceptually simple colour spaces, such as Hue-Saturation-Intensity, could also be used in determination of turbidity.

8.2.3 Survey Location

The survey attempted to cover an area 10km upstream of the River Taw and Torridge from their point of confluence. This area of North Devon is a popular tourist and holiday location. The recreational facilities in the area include all forms of water sports, fishing and walking, with sandy beaches and rocky cliffs along most of the coastline.

Figure 8.12 shows the study area, with the towns of Bideford, Barnstaple, Appledore, Westward Ho!, Braunton and Northam. Also shown are the positions of the proposed outfalls.

8.2.4 The Survey

The spring tides occurred on the 30th,31st of July and the 1st of August, for these dates high water, was at 7.30, 8.10 and 8.54 in the morning. The water quality survey was being carried out by Wimpol from 1 hour before to 1 hour after high water. The problems of getting the plane to the area for this time made the spring tide survey impossible, and it was therefore decided to opt for the neap tides on the following weekend, the 6th, 7th and 8th August. For these dates high water was at 13.15, 14.30 and 15.45 in the afternoon. This gave three possible dates to wait for good weather.

On the Friday evening the modified door was fitted to the Cessna at Coventry airport. The weather that weekend was clearest in months, and the survey was flown on the Saturday (6th August).

The cameras were fitted to the door and tested at Swansea airport, approximately 25 minutes across the Bristol Channel from the Taw-Torridge estuary.

The testing and installation of the cameras involved the setting of the camera lenses to give equivalent ground coverage, and fitting a visible light blocking filter to the black and white camera. With the filter in place the camera had an extremely limited range of contrast, moving the aperture ring a millimetre either side of the optimum saturated the image.

To ensure sufficient contrast across the small field of view obtained, about 200m at 3000ft altitude, the automatic gradient control was used. This made subsequent mosaicing of more then a few adjacent scenes difficult.

Using the video monitor the contrast of both camera systems was checked and found sufficient. Unfortunately the video tape recorder used to acquire the near infrared imagery would not allow preview of the image being recorded. Vibration caused the aperture ring of the BW camera to open and consequently land scenes acquired with the near infrared system were saturated. The arrival over RAF Chivenor at about 1.00 pm coincided with high water and the tape recorders started. A reconnaissance pass at 1000ft was first undertaken in an attempt to find Wimpol's survey boats. This in itself was a major problem with the area a hive of water sports activities.

On the three different days of the survey the boats were due to be surveying in different areas of the estuary. On the 6th August this was in a line between Weare Giffard bridge and the dismantled railway bridge near Hallsannery, and a line between old Bideford bridge and Instow Sands (see Figure 8.13).

With the area reconnoitred, the survey was continued at 3000ft. The problem of co-ordinating flight lines was immediately apparent. An initial attempt to fly in lines along each bank of the river, became a torturous and nauseating route upstream. Subsequently visible landmarks were used in an attempt to fly in relatively straight lines along each estuary. In estuarine and coastal areas the choice of landmark becomes severely limited. Major coverage was given to the Torridge Estuary followed by a number of passes along the confluence of the estuaries and up the Taw into Barnstaple. Figure 8.14 shows the approximate position of the flight path.

8.2.5 Processing

The processing of the data was carried out in the weeks following the survey. In the processing two aims were identified, firstly to geometrically and radiometrically correct the strips of data, and secondly to analyse the water colour to give some measure of turbidity.

The various stages of the process are shown in Figure 8.15 and described in the following sections.

8.2.5.1 Decoding and Digitisation

To reduce the storage requirements for the volume of data acquired, each frame was digitised at a resolution of 765 x 576 and then the centre 512 x 512 block was reduced to 256 x 256 pixels for storage. Each stored frame therefore occupied three 64KB disk files. This stored image consists of data from only one of the interlaced fields, and is therefore free of problems caused by non-registration of fields. The effective pixel resolution of the image was therefore reduced from 0.35m to 0.7m.

At this stage processing of the infrared imagery was abandoned due to saturation of the land scenes. The red, green and blue components of the colour imagery were decoded and digitised, producing a three band multi-spectral data set.

8.2.5.2 Preprocessing

After experimenting with Fourier filtering, noise was removed from the imagery with simple meanal convolution operations. A meanal filter replaces each pixel by the mean of itself and its eight neighbours. As the data was to be eventually reduced in resolution down to 2m, to geocode to a 1 km frame, this procedure could be performed without any loss in image definition.

8.2.5.3 Production of Geocoded Imagery

To geocode the data topographic features, such as roads, rivers, field boundaries and large buildings, were digitised from the OS 1:25000 sheets for the area. This process is described in Chapters 4 and 7. Individual 1 km squares of data were extracted, stored in a vector database and used as a base for correcting the imagery.

Geometric correction of the data has been described in Chapter 6. This involved the interactive placement of frames, reduced to 2m resolution, on the 1 km base. Placement of the frames used a simple 2D transformation process involving components of rotation and scaling.

Due to the product of errors the generation of a seamless mosaic proved impossible over more than a few frames. The simplest mosaicing procedures use the regression of pixel brightness in an area of overlap to derive a transformation for one of the frames. In mosaicing a strip of frames, the overlap was generally about 20%. If the process moves along the flight path, the brightness transformations sum for each new frame. This can result in a trend of increasing or decreasing brightness along the strip.

Mosaicing problems were caused by the use of the auto-gradient control function built into the camera. The simplest solution would have been to disable this control prior to the survey. More complicated solutions involve colour matching adjacent scenes using colour space measurement, alternatively it could be suggested that the gradient control alters the components of recorded colour equally and hence is analogous to variable lighting conditions. As turbidity can be related to colour hue and saturation there may be no requirement to mosaic scenes in production of a turbidity map.

8.2.5.4 Turbidity Measurement

A number of frames were analysed to assess the feasibility of extracting turbidity by various methods. The frames selected corresponded to one of the survey sites of the Wimpol survey at the disused railway bridge. Figure 8.16 shows the location of the scenes processed.

The procedure for derivation of turbidity is detailed in Figure 8.17. The first stage is to mask the water so that the land mass is ignored in the analysis. This mask can be derived manually by tracing the water boundary, alternatively the mask can be extracted from the spectral characteristics of the water. An infrared image would be particularly useful for this process.

The next stage in the process was to produce two measures of the variation of turbidity in the water. The first measure was taken as chromaticity x, see Section 8.2.2. This was derived from the measured brightness variation as there was no calibration data relating digital number to radiance. The second measure was hue derived from a Hue-Saturation-Intensity transformation. This transformation is described in numerous remote sensing and image processing texts.

The frames were processed individually so that the independent measures of turbidity, hue and chromaticity x, could be correlated in the areas of overlap.

8.2.6 Conclusions

Vibration during the flight caused the lens aperture on the infrared camera system to open, saturating the image of all land areas. Over water imagery was obtained; however this consisted almost entirely of sunglint, seagulls and watersports activities.

The lens field of view was far less than expected, 200m. A more suitable swath width for this application would have been about 500m. This made the task of piecing together strips of the survey and the problem of radiometric correction and mosaicing more complex.

It is clear from Figure 8.19 that the process of derivation of turbidity does not produce consistent results across adjacent frames. This is due to the major assumption that there is a monotonic brightness increase between frames introduced by the auto-gain mechanism of the colour camera. In addition to the auto-gain, the camera had an automatic red-blue shift compensator. The effect of this automatic mechanism was not visible during acquisition of the imagery due the use of a monochrome monitor, but can be observed in the colour shift between frames.

In terms of an initial feasibility study, the survey highlighted a number of important improvements that needed to be made to the survey system. These improvements would reduce the radiometric errors present in the survey data and aid interpretation of the imagery.

1. A wider swath width would provide a more synoptic viewpoint and would facilitate geometric correction.

2. A multiple camera array system would reduce the radiometric errors introduced by encoding and decoding of the data, and with manual control of colour shift maintain the colour range throughout the survey.

3. The disabling of all automatic controls, aperture, gain and offset would enable easier mosaicing of frames.

With these improvements variation of colour hue and image brightness would remain roughly constant throughout the survey enabling turbidity and variable lighting to be quantified and correlated.

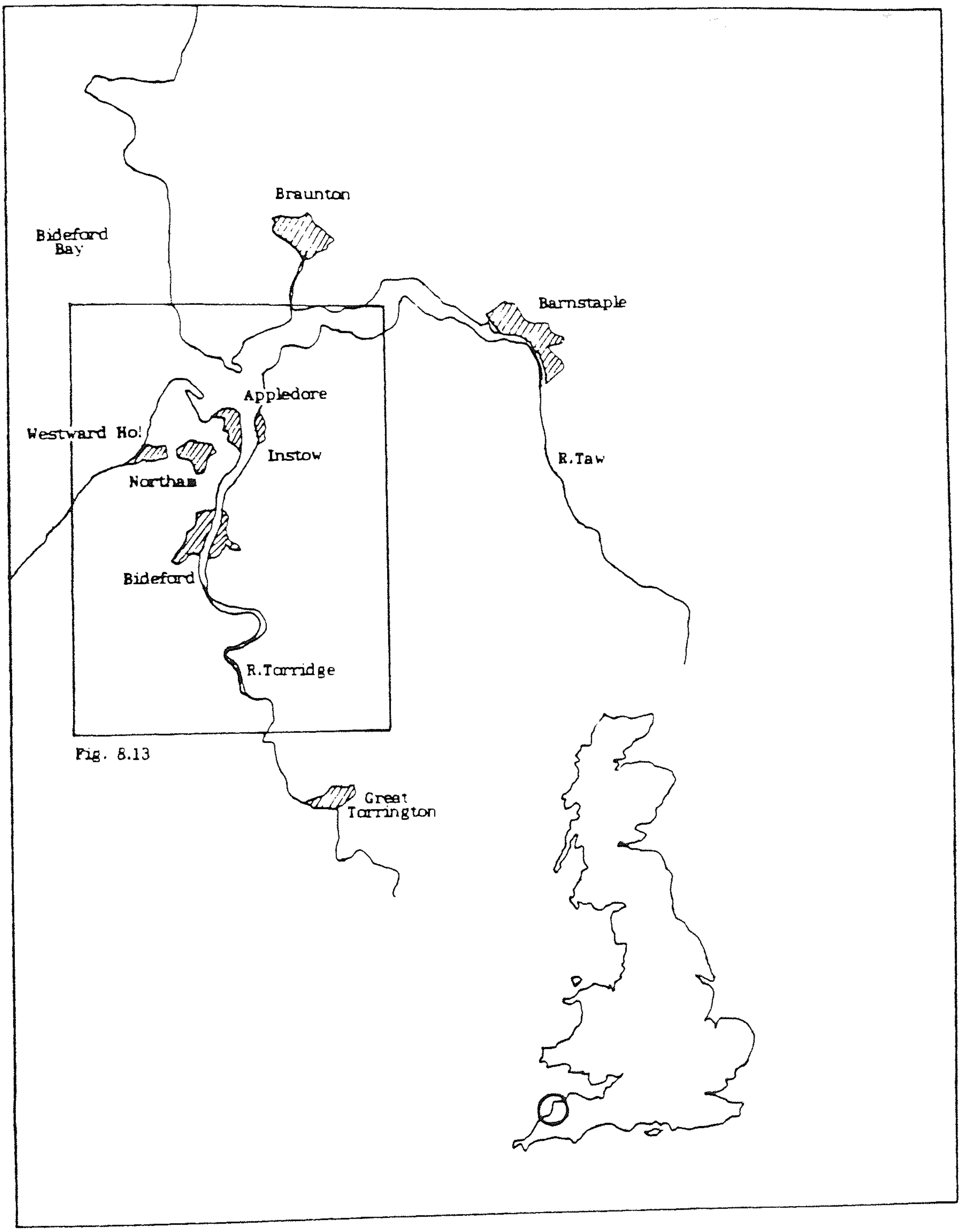


Figure 8.12. The Taw-Torridge Estuary.

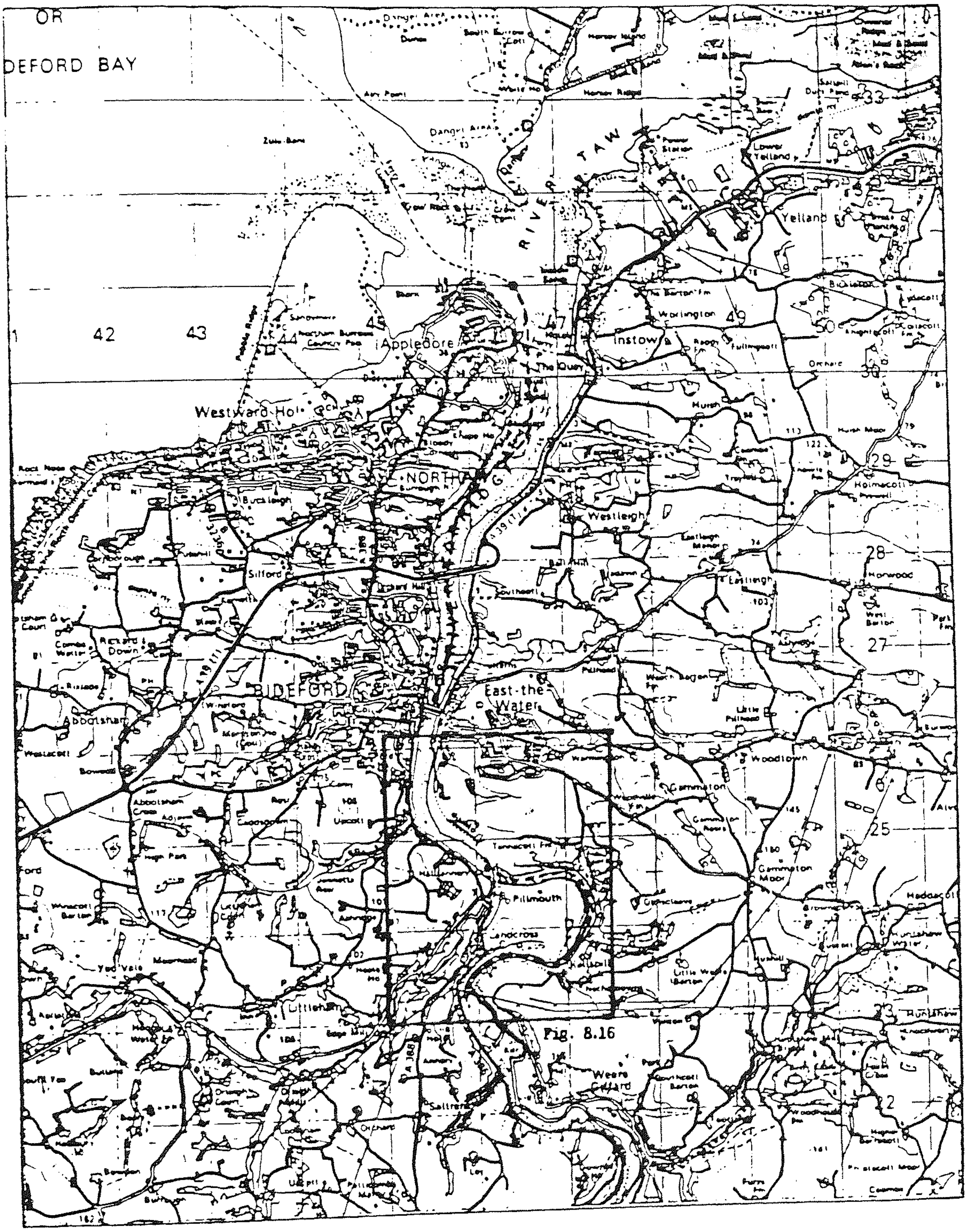


Figure 8.13. Wimpol Survey Locations.

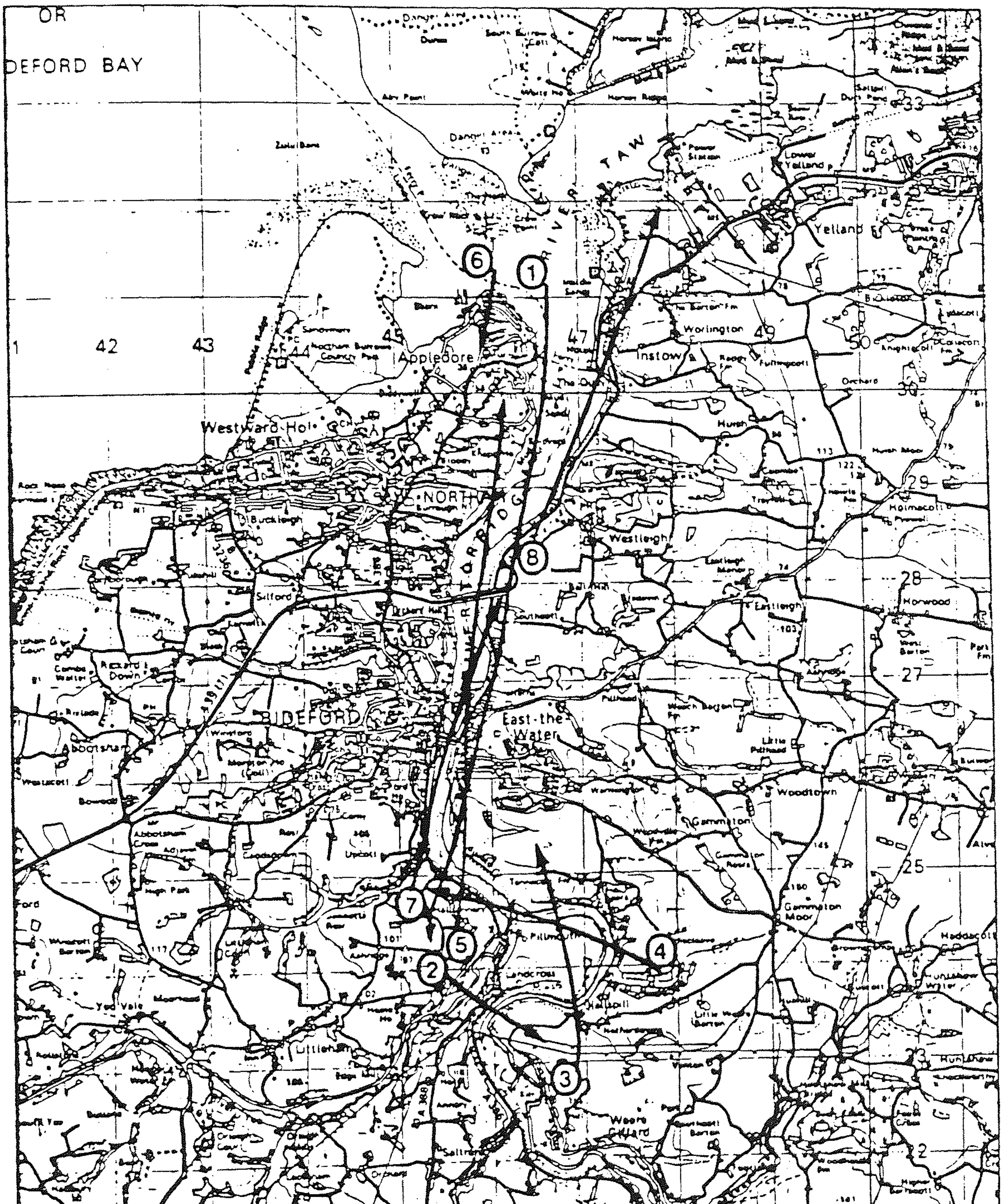


Figure 8.14. Flight Path Location.

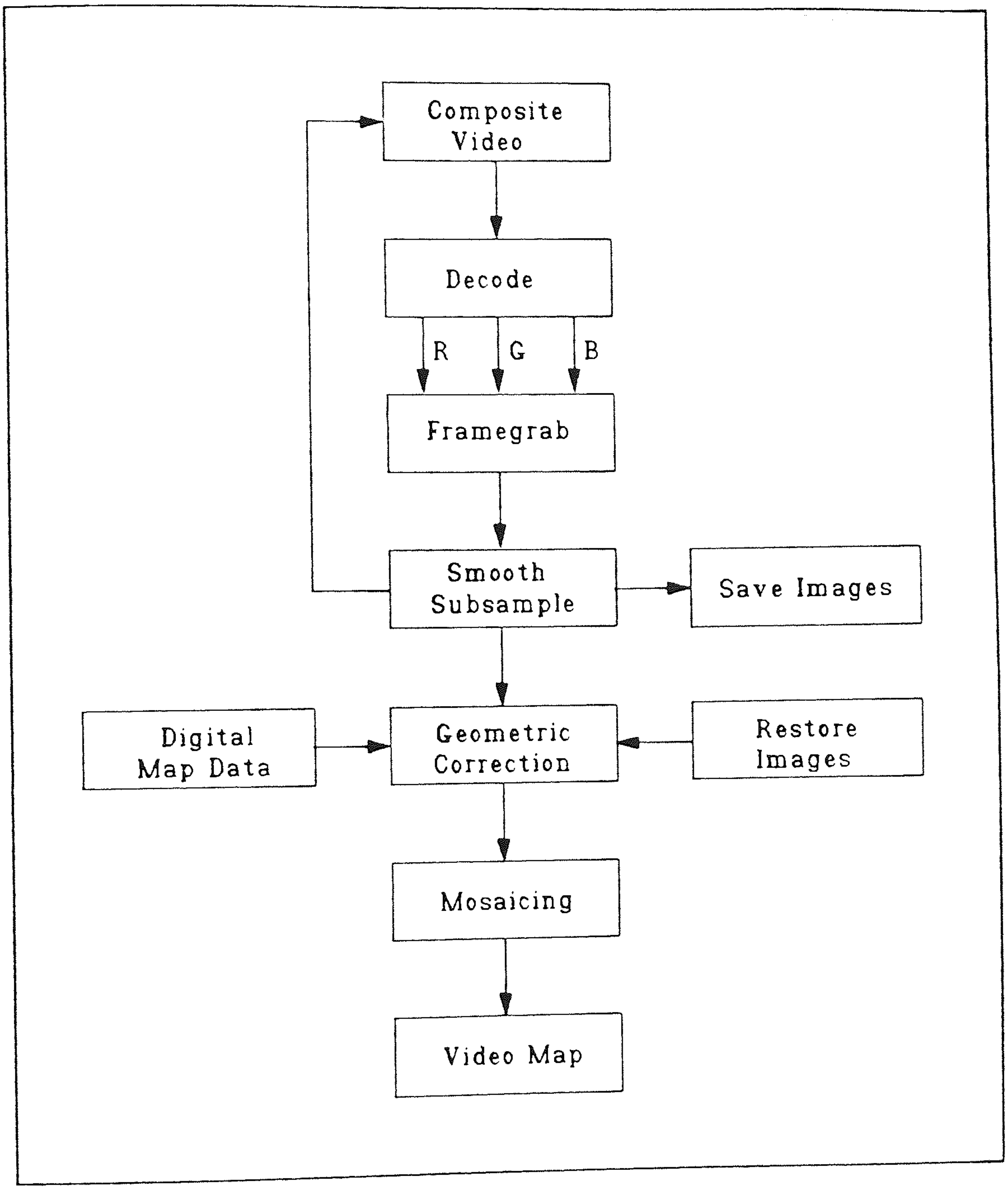


Figure 8.15. Processing the Taw Survey.

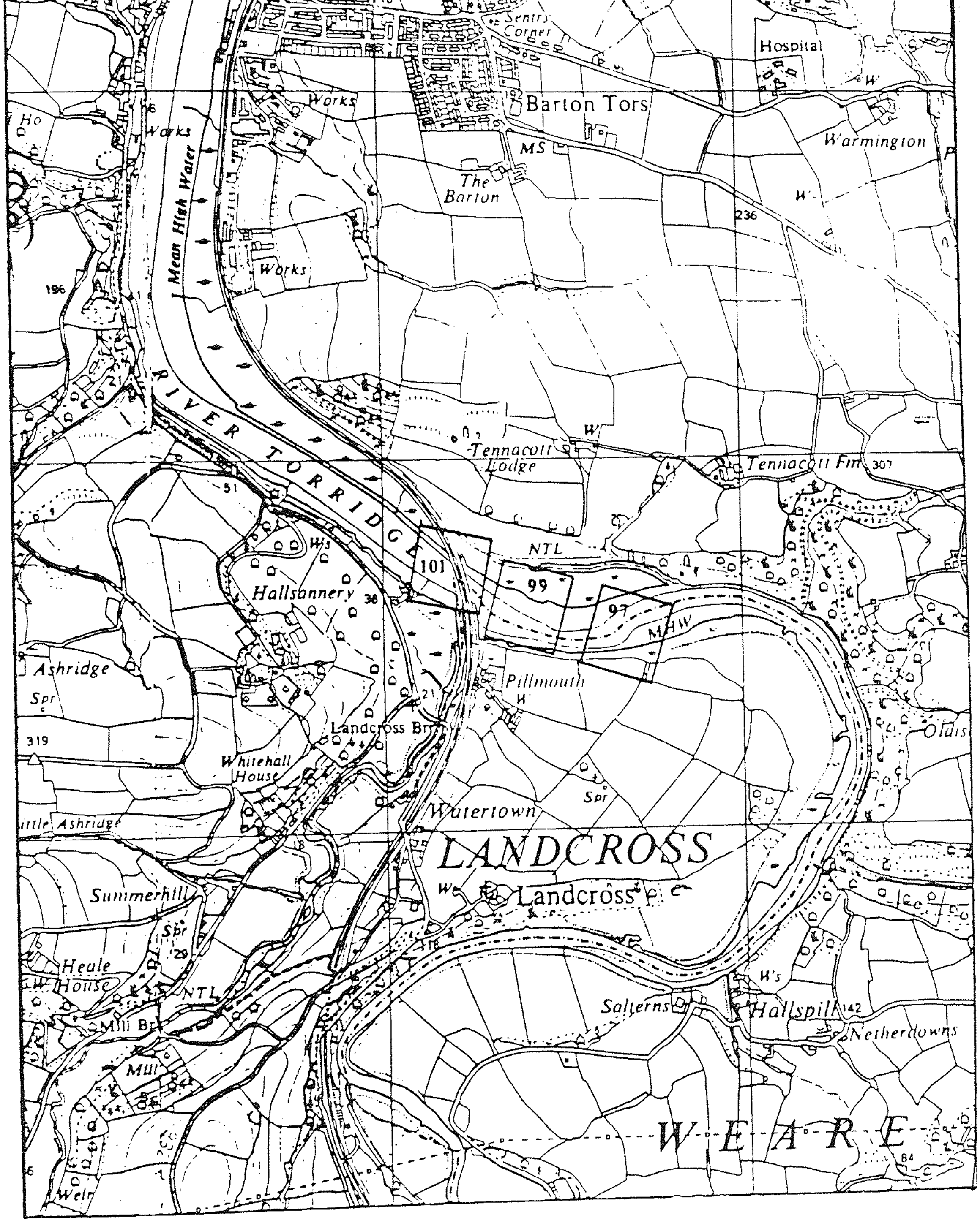


Figure 8.16. Location of Processed Scenes.

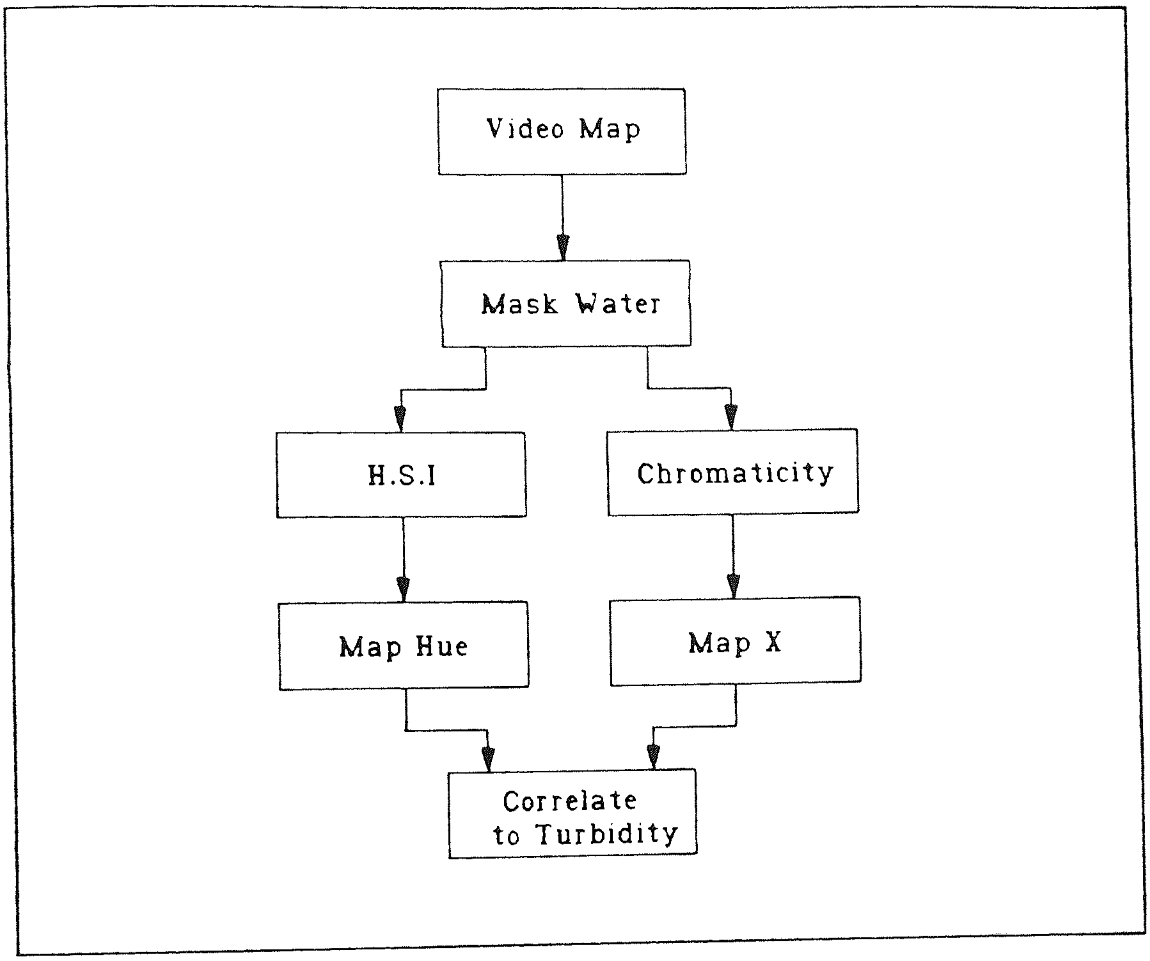


Figure 8.17. Turbidity Extraction Process.

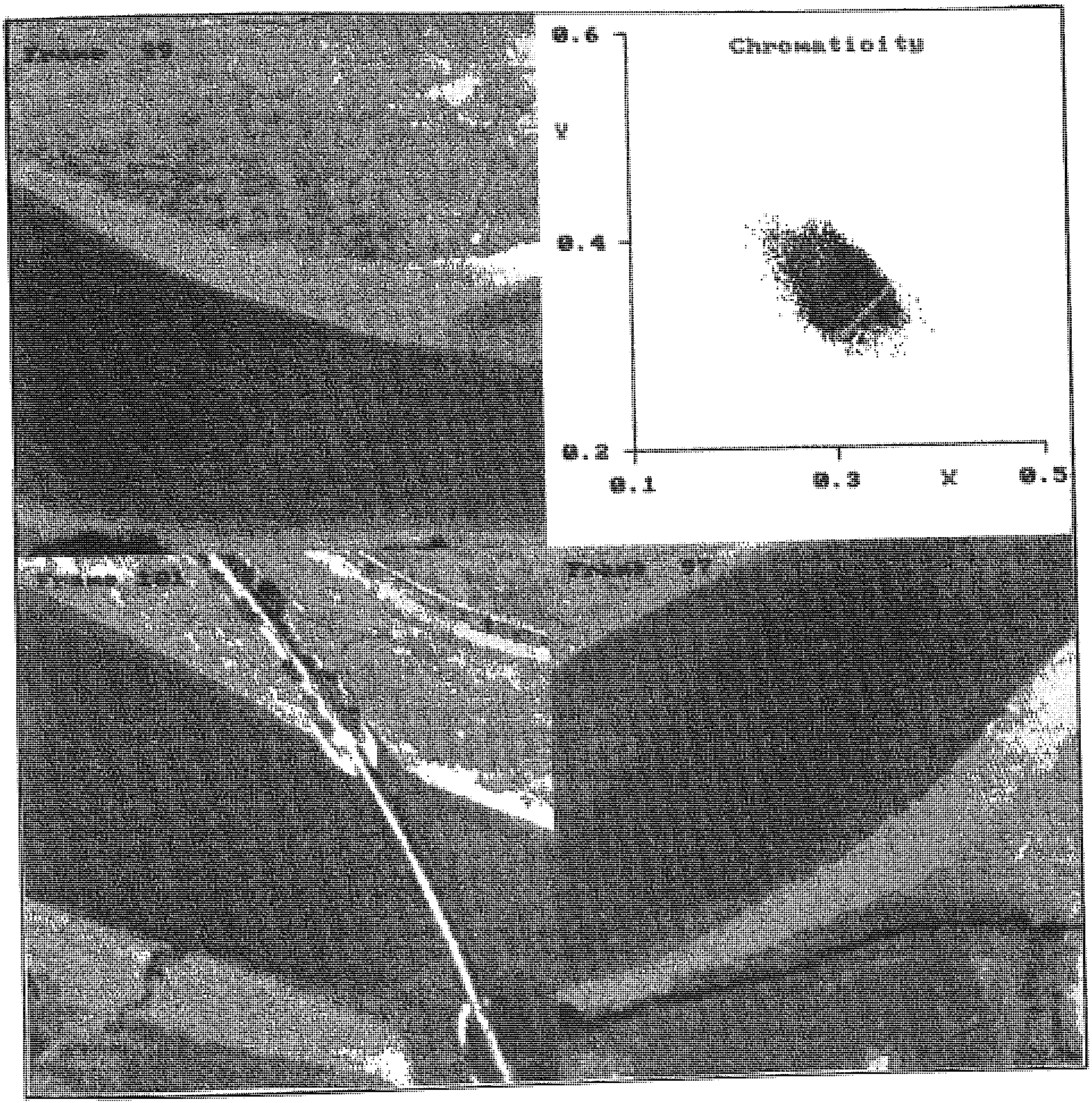


Figure 8.18. Chromaticity Variation.

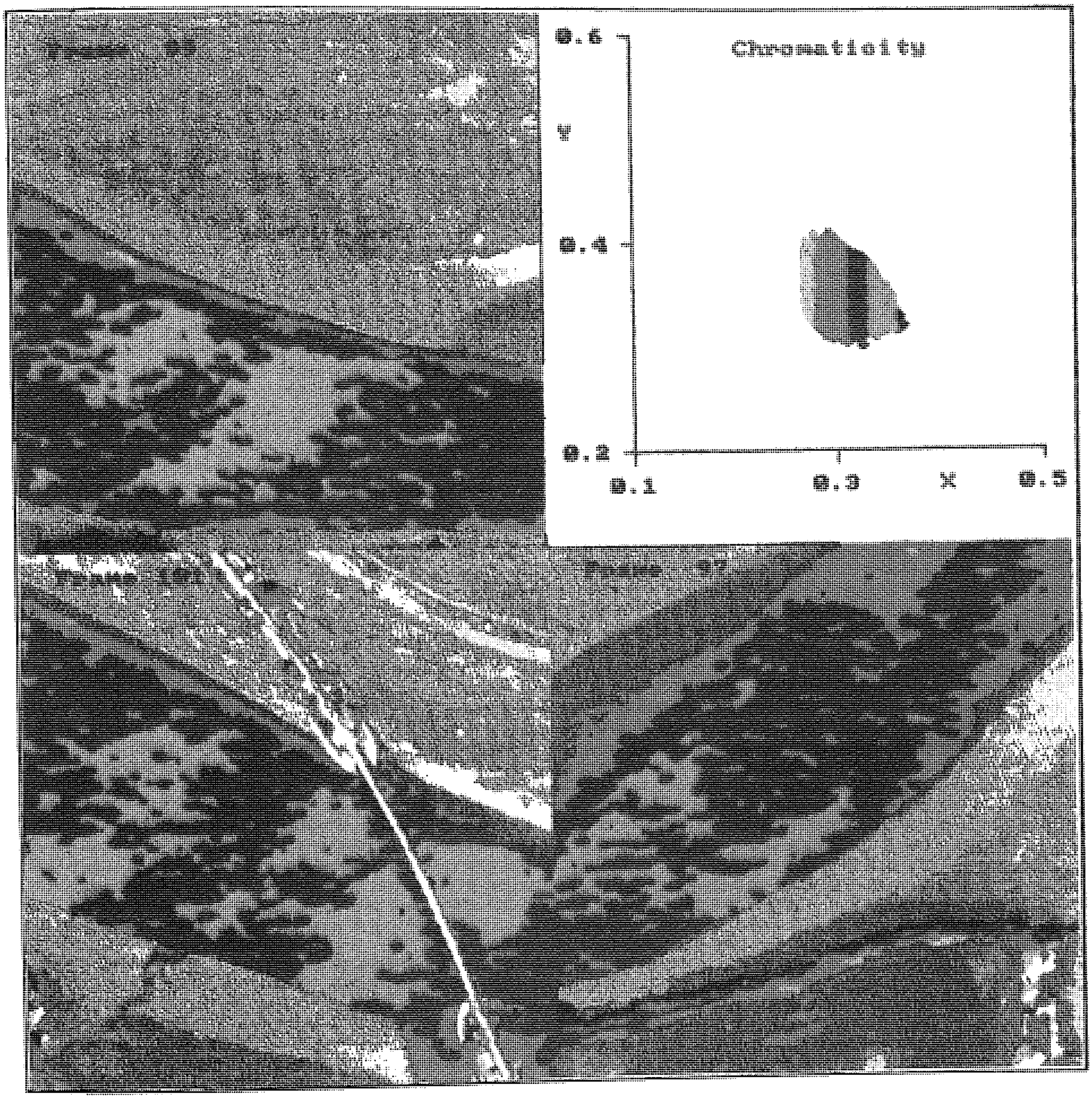


Figure 8.19. Turbidity Map.

8.3 Land Use Mapping of Water Protection Zones

This study was an application of remote sensing for land use classification, and formed the basis of an initial feasibility study, undertaken for the Severn Trent Water Authority, to assess remote sensing as a tool to map potentially hazardous non-point source groundwater pollution by chemical fertilizer application.

The first part of the study used the Landsat TM satellite imagery to map land use over a sandstone aquifer outcrop. For this scale of regional project, such as within a large outcrop, satellite imagery provides the most suitable and cost effective method of mapping land use and identifying potential pollution hazard.

8.3.1 Introduction

The problems of groundwater contamination from fertiliser sources has become more significant in the light of the European Community (EC) rulings on water quality limits. It is predicted that at present about 4 million people in Britain drink water that is above the EC nitrates limit. This number is expected to rise as nitrates build up in the groundwater.

Since the 1974 Control of Pollution Act government ministers have had the power to set up water protection zones, in which the use of nitrates and other pollutants would be regulated. The Ministry of Agriculture opposed these powers and consequently they have never been invoked. The result is a lack of control over intensive farming practices. In the light of the privatisation of the water industry, Missing page(s) from the bound copy

8.3.2.1 Location

The study area is the outcrop of the Triassic aquifer, extending some 15 km east-west by 70 km north-south, situated to the north of Nottingham. Figure 8.20 shows the location of the outcrop.

The area is mainly arable, with the dominant land use being cereal cultivation, subject to both spring and winter planting. Oil seed rape, main crop potatoes and sugar beet are also prominent along with a small percentage of legumes and vegetables.

8.3.2.2 Land Use Classification

The first stage of the project used a Landsat TM image acquired for the 26th April 1984. This is not a particularly suitable month to assess crops, an image acquired later in the growing season would make discrimination of crop types easier. Given the 16 day interval between satellite overpasses and the extent of cloud cover in these periods, it proved impossible to acquire a new scene for July or August 1988.

8.3.2.2.1 Fieldwork

The objectives of the fieldwork was to identify from farm and parish records at least ten sites of each land use class identified by Severn Trent.

These classes were:

1. Cut grass, grazed grass, rough pasture
2. Winter cereals
3. Spring cereals
4. Root crops
5. Oil seed rapes
6. Legumes
7. Other crops
8. Forest
9. Urban
10. Bare soil
11. Gravel pits

This was achieved except for class 6 where only 3 fields planted to legumes in 1984 were found. The similarity in growing season of root crops and legumes, together with their appearance on the 1984 imagery suggested their aggregation into a single class. Further experience during the fieldwork stage and on examination of the image data led to several revisions of the land use classes to the following:

1. Ley grassland
2. Winter wheat
3. Winter barley
4. Oil seed rape
5. Spring barley
6. Root crops and legumes
7. Permanent grassland
8. Deciduous woodland
9. Coniferous woodland
10. Urban areas
11. Quarries
12. Water
13. Reclaimed mining land
14. Recreational Land

8.3.2.2.2 Image Processing

A subscene covering the outcrop area was first extracted. This was 1024 x 3072 pixels in size, which with the 6 bands of data used, occupied some 18 MB of storage.

The first stage in production of a land use classification is to derive statistical measures relating to the spectral properties of each class in each band. This is done by marking areas on the image display and scanning the image data for pixels within each area. Each area is known as a training area.

Half of the identified sites were used as training areas, the remaining half were used to assess the accuracy of the classification procedure.

The classification performed was a maximum likelihood classification. This algorithm works by determining a probability surface for each class in each band. Each pixel is then assigned to the class for which the probability for inclusion is the highest.

The accuracy of the classification is presented in what is known as a confusion matrix. In this matrix the number of pixels in the training areas correctly classified are represented along the leading diagonal. Off diagonal elements represent the misclassified pixels, either incorrectly omitted from the class or incorrectly included from another class. The overall accuracy is then calculated by summing the leading diagonal and dividing by the total number of pixels tested.

Number of Pixels:

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pixels	170	195	149	140	177	139	90	127	150	198	75	107	67	159

Table 8.1 Number of Pixels used for Test on Classification Accuracy.

Confusion Matrix:

		Commission													
	Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Class														
	1	170	1	0	0	0	0	16	6	0	0	0	0	0	2
	2	0	194	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	149	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	140	0	0	0	0	0	0	0	0	0	0
O	5	0	0	0	0	126	0	0	4	0	0	0	0	0	0
m	6	0	0	0	0	51	139	0	0	0	0	2	0	0	0
i	7	0	0	0	0	0	0	36	0	0	0	0	0	32	0
s	8	0	0	0	0	0	0	11	100	0	72	21	0	0	47
s	9	0	0	0	0	0	0	0	0	150	3	0	0	0	0
i	10	0	0	0	0	0	0	0	0	0	123	2	0	0	0
o	11	0	0	0	0	0	0	0	0	0	0	50	0	0	0
n	12	0	0	0	0	0	0	0	0	0	0	0	107	0	0
	13	0	0	0	0	0	0	7	16	0	0	0	0	35	0
	14	0	0	0	0	0	0	20	1	0	0	0	0	0	110

Overall Accuracy: 83.8%

Table 8.2 The Classification Confusion Matrix.

From the confusion matrix it can be seen that 5 classes show little or no overlap with other classes, these are:

2. Winter wheat
3. Winter barley
4. Oil seed rape
9. Coniferous woodland
12. Water

Two further classes were identified with high accuracy, giving errors only in commission (inclusion of pixels in other classes):

1. Ley grassland
6. Roots and legumes

There is a certain amount of confusion between the remaining classes with permanent grassland, class 7, being least successfully classified. In this particular case the majority of the permanent grassland has been classified as other grassland types, and most significantly ley grassland. This confusion could be a source of error considering the comparative quantities of fertilizer applied to these classes.

Once completed the land use classification was geometrically corrected to ground control points. This involves a least squares fit to derive the 2D transformation equations. The accuracy determined for this process gave a root mean square error in positional accuracy of 0.97, corresponding to 29m.

8.3.2.3 Outcrop Delineation

To complete the project the location of the aquifer outcrop had to be overlayed on the land use classification.

This was achieved by tracing the outcrop from the geological and administrative boundaries provided. This was video digitised and the boundary vectorised at an effective resolution of 25m, in 6 figure OS grid co-ordinates.

The 1024 x 3072 pixel classified land use map was then sectorised into 512 x 512 pixel extracts. Each extract was overlayed with the boundary data, creating a mask within which the number of pixels in each class was counted.

This was totalled to give the areas of each class on the outcrop, Table 8.3.

Each extract was also overlayed with the OS grid prior to printing out at a scale of 1:100,000. Figure 8.21 shows an extract of the classified image overlayed with the outcrop boundary and grid data.

Class	Type	Area (Ha.)	% Area
1	Ley grassland	12494.4	15.83
2	Winter wheat	5373.0	6.81
3	Winter barley	11284.6	14.30
4	Oil seed rape	1731.7	2.19
5	Spring barley	7915.8	10.03
6	Roots & legumes	9023.4	11.43
7	Permanent grassland	1847.3	2.34
8	Deciduous woodland	13496.2	17.10
9	Coniferous woodland	3695.0	4.68
10	Urban areas	9310.3	11.80
11	Quarries	453.1	0.57
12	Water	480.7	0.61
13	Reclaimed land	1089.6	1.38
14	Recreational land	738.0	0.93
	Total	78933.1	

Table 8.3 Area of each Land Use Class over Outcrop.

8.3.3 Conclusion

Although using an April image, the classification of the satellite imagery proved fairly successful in deriving land use over the aquifer. Using an image acquired during the main growing season may help to resolve some of the confusion present between classes, although the necessity for this may well also depend on the local land management practises concerning nitrate fertiliser application. These practices tend to vary considerably, although several general points have been noted by the Ministry of Agriculture. Winter cereals and oil seed rape require the largest amounts of fertiliser due to their longer growing seasons. Similarly, ley grassland receives nitrogen according to the number of cuts per year and may range from 120 Kg/ha/year to 300 Kg/ha/year. Permanent grassland in general receives less nitrogen than ley grassland.

Nitrogen from fertiliser sources would be expected to enter the ground as a result of leaching by recharge to the aquifer. The relative amount of leaching from each type of crop is difficult to assess. The Ministry of Agriculture regard deep rooted crops as less of a problem, shallow rooted crops such as potatoes are cited as particular cause for concern, as is high stocked pasture land.

Perhaps of greatest significance to the quantities of nitrate pollution is the temporal distribution of rainfall relative to fertiliser application - a phenomenon not studied in this project. Clearly a greater pollution risk occurs if there is heavy rainfall shortly after application. This could only be effectively studied by monitoring fertiliser application groundwater, flow and rainfall over a large period of time.

In conclusion, it would appear that multi-spectral classification techniques combined with integrated cartographic data can provide useful information in estimating probable supply of nitrogen contaminated recharge to an aquifer. The resolution of the satellite imagery may well be found insufficient for all but regional studies. Further discussion with Severn Trent Water has opened the possibility of using airborne multi-spectral video for land use classification of protection zones local to groundwater pumping wells.

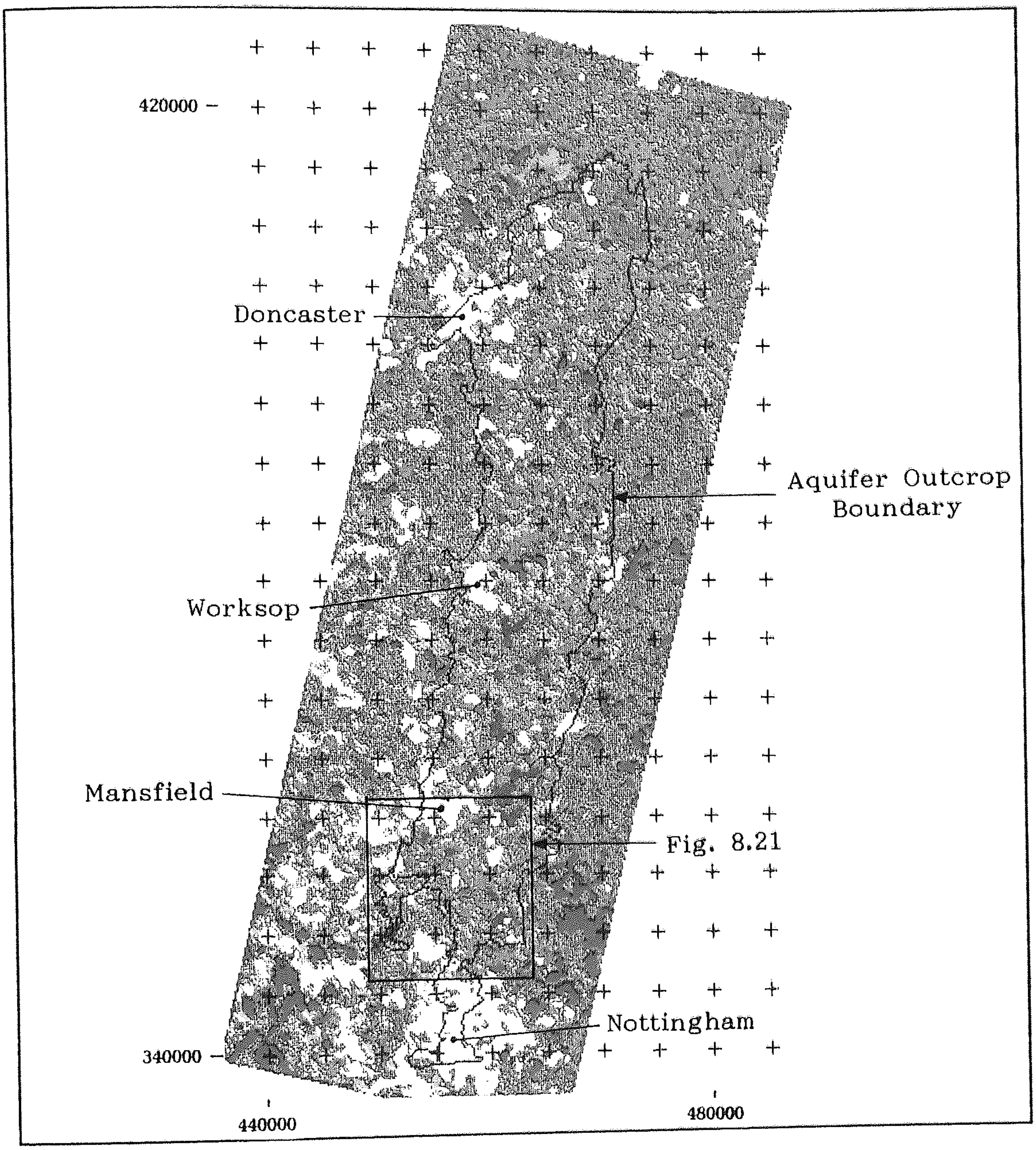


Figure 8.20. Aquifer Outcrop Location.

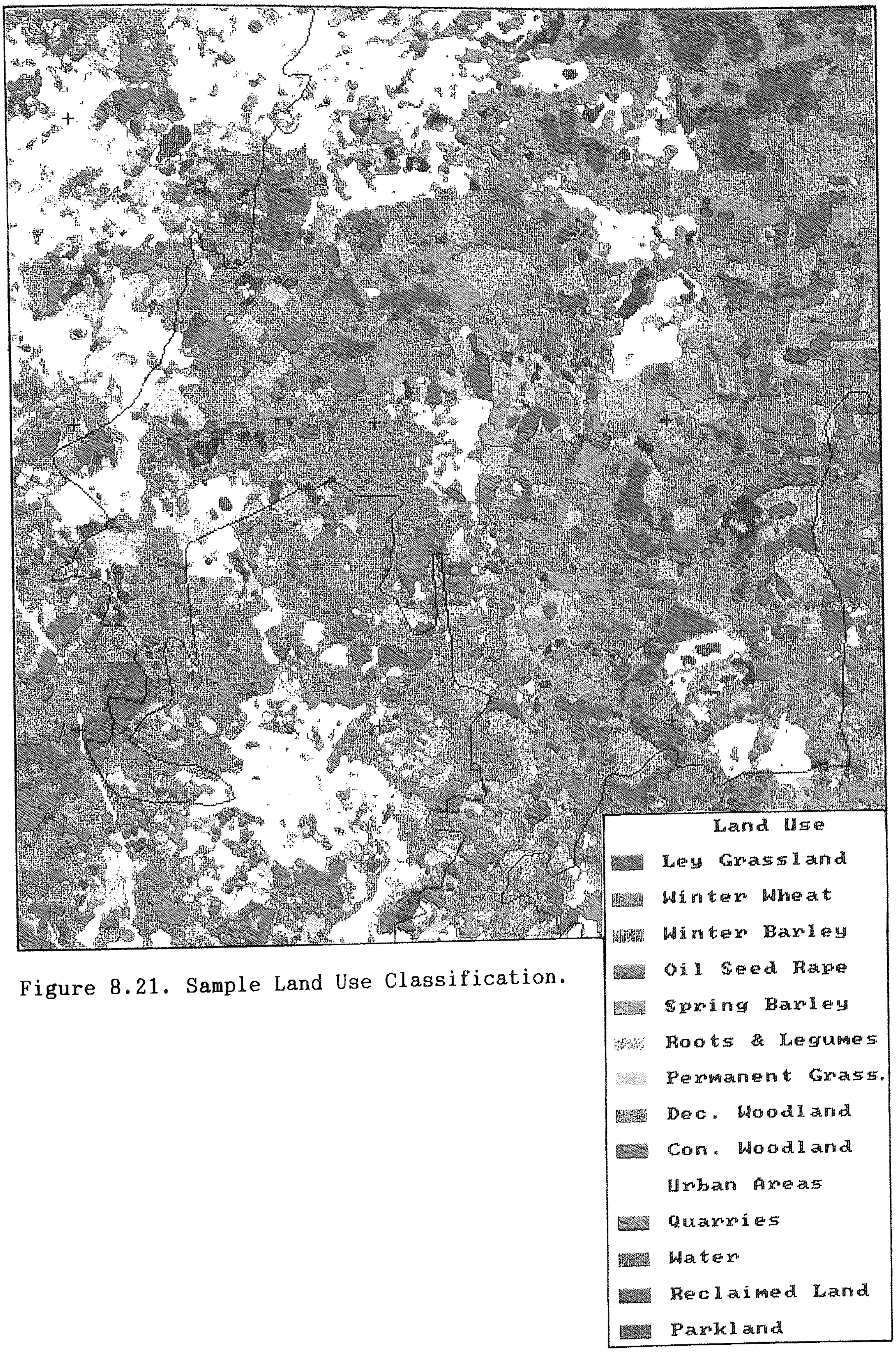


Figure 8.21. Sample Land Use Classification.

8.4 Magat River Basin Erosion Study

This application was carried out in collaboration with Sue White from the Overseas Development Unit of Hydraulics Research Ltd, Wallingford, Oxfordshire. This study forms part of a larger ODU project on the Magat river catchment, situated in Central Luzon, Philippines. The study involved creation of a spatial database to be used in erosion risk modelling, and was an attempt to improve existing predictive methods such as lumped catchment and plot scale models.

8.4.1 Introduction

Soil erosion is a major problem in all developing countries. Pressure for increased food production leads to agricultural development in marginal lands, often causing accelerated soil erosion rates and high sediment loads in rivers and canals. The net result of this is a reduced flow capacity due to siltation, loss of available irrigation water and reduced life of reservoirs.

The prediction of soil erosion is largely qualitative and is not normally considered in river basin management until the erosion problem is already well advanced. Accurate predictions of erosion and sediment yield are therefore needed to make economically sound planning decisions. Most predictions in this field have a reputation of being inaccurate, in cases predicted sediment yield has been as low as 10% of that measured, and are therefore largely ignored at the feasibility stage of planning.

Existing predictions are either based on models developed for small field areas, such as the Universal Soil Loss Equation described by Wischmeier and Smith (1978), or on lumped catchment models such as Fournier (1960) and Fleming (1960). These models, in common with most other hydrological models, suffer from the effect of scale. Distributed models, developed on a grid or sub-catchment basis, such as developed by Beasley (1977) and Fleming and Walker (1976), offer greater detail within the catchment, but use large amounts of data. The spatial database can be seen as a means of storing data for large scale distributed models.

Clearly some improvement on existing predictive methods is required. The overall aim of the project was to develop improved methods for predicting erosion risk and sediment yield using remote sensing and Geographic Information Systems technology. This will then be calibrated on the basis of sub-catchments, a series of nested catchments and the catchment as a whole.

8.4.2 Location and History

The Magat river catchment is situated between 16°N and 17° N, and between 120° 50'E and 121° 30' E, in central Luzon, Philippines. Figure 8.22 shows the approximate location of the catchment.

The Magat reservoir was impounded in December 1982 and drains an area of 4123 km^2. The reservoir has a capacity of approximately 1080 x 10^6 m^3, with a live storage of 810 x 10^6 m^3. The area is prone to tropical cyclones and high intensity rainfall.

Throughout the area there is intense pressure for agricultural land and illegal logging operations, this has led to extensive deforestation, and there is consequently a high risk of erosion and high sediment yields in the rivers. In an initial study, over the first 21 months since impoundment, deposition had occurred at a rate twice that predicted in the feasibility study. This will reduce the operating life of the reservoir from 100 to 25 years.

Recent river basin management policies are attempting to reduce the sediment yield by undertaking soil conservation and re-forestation projects. This will be monitored over the next few years, providing insight into the effectiveness of these management policies. Due to the quantities of sediment already in the drainage system, it is probable that the effect of these works, in terms of reduced sediment yield, will not be significant for a number of years.

8.4.3 Database Production

The first stage in production of the database was to assess the available information, and the necessary requirements of an erosion risk/sediment yield model for the area.

These included, topography, slope, aspect, geology, soils, land-use, land management units, rainfall, stream network, sub-catchment boundaries and erosional competence. The location of several topographic and hydrologic features was also required for reference.

The methods of production of these databases is detailed in Chapters 4, 5 and 7. Figure 8.23 shows the relationships of the various components of the database.

8.4.3.1 The Co-ordinate system

The majority of maps and charts used for input were plotted on the Universal Transverse Mercator projection. All the maps were in different scales ranging from 1:500,000 to 1:50,000.

Rather than use the UTM projection in the database, a local rectangular projection system was used, giving 12 pixels/minute of latitude and longitude. This gave rectangular database pixels of 150.8m in the x and 154.9m in the y direction. The advantage of this was that it enabled the entire catchment to fit onto the 768 x 768 pixel framestore and provided a basis for co-registration of the various maps and charts.

8.4.3.2 Class Maps

There were a number of class maps produced these included Geology, Soils/Land Management Units, Land Use and Sub-catchments.

The Geology and Soils/Land Management Units were produced using a process of video digitisation, geometric correction, boundary extraction and seed fill. For both the brightness values were grouped so that both a colour and monochrome representation were possible. Figures 8.24 and 8.25 show the Geology and Soil Maps produced.

The sub-catchment areas were produced using a scan digitised boundary map. The raster representation of the boundaries was geometrically corrected and the sub-catchments classified using a seed fill. For this database the boundaries did not have to be vectorised, due to the radiometric fidelity of the scanning system. After the seed fill process the boundaries are removed by a median filter operation, where the pixels on the boundary lines are replaced by the median of the surrounding pixels. Figure 8.26 shows the sub-catchment map produced.

The final class map involved the use of remotely sensed data. For this study a Landsat MSS image was acquired for 17th March 1987, and supplied on CCT (computer compatible tape) by the ground receiving station in Bangkok, Thailand.

This was processed and a subscene 1536 x 1536 pixels was extracted from the CCT. The Landsat MSS data provides 4 bands of data at a resolution of 56 x 79m.

Hydraulics Research personnel had visited the area and could provide ground truth data, enabling certain land use types to be identified by their spectral properties in the 4 MSS wavebands. This formed the basis of a multi-spectral supervised classification. This process was carried out on the extracted subscene and a land use map produced.

The next stage in the process was to correct the satellite derived land use map to the database co-ordinate system, using topographic data visible on both the map and satellite image. The satellite image and land use classification derived are shown in Figures 8.27 and 8.28.

The accuracy of the land use classification produced was questionable, but was certainly more accurate that any existing land use map. With a large part of the upland areas of the catchment under the effective control of guerrillas, it would also prove extremely difficult to prove the accuracy of the classification. Its application in the erosion hazard model depends on the determination of the interception of rainwater by various land use classes, taking into account plant canopy and litter. This is entirely empirical and most likely a much greater source of error than that introduced by misclassification.

8.4.3.3 Surface Maps

In addition to the class maps, there were a number of surface maps required. These surface maps are not collection of areas of one class but are general brightness surfaces depicting height, slope, aspect and rainfall.

Of these, the most time consuming to produce was the digital elevation model or height surface map. The production of this map involved the raster digitisation of a topographic map. In this study the only available topographic map was a 1:500,000 Tactical Pilotage Map (TPM). This map was digitised on a drum scanning densitometer, with the aim of increasing the ability to read and follow the contours on the map. The contours and spot heights were digitised from sectors extracted from the TPM image data, so that digitisation could be carried out at the highest possible resolution.

Once completed the data was recalled onto the database co-ordinate system and an interpolation performed as described in Chapter 5. A variety of different algorithms were used. The best result, with the smoothest surface, was achieved by a linear least squares interpolation to a coarse grid, followed by a bi-cubic interpolation from the coarse grid to the fine grid, equivalent to the database co-ordinate system. Figure 8.29 shows a contour map derived from the Digital Elevation Model (DEM) by colour density slicing.

The DEM is used to generate slope and aspect maps. For the purposes of this study slope was determined as the maximum slope from the centre to the edges of a 3 x 3 local surface. Aspect was derived as the direction of the maximum slope. Slope and Aspect maps thus derived are represented by Figures 8.30 and 8.31.

The average rainfall surface was generated in much the same way as the DEM from a contoured rainfall map. These contours corresponded to isohyets, with spot heights representing raingauge measurements. This database then gives an average rainfall intensity for each pixel of the catchment database. Figure 8.32 shows the rainfall map, coloured by ranges of intensity.

8.4.4 Conclusion

The production of this spatial database allows manipulation of the data in a spatially distributed form. Sediment Yield/Erosion Risk can now be calculated at every level from a pixel basis, through sub-catchments, nested catchments to the total catchment. Several models have already been attempted based on the SLEMSA model developed for the SADCC region, Stocking (1987). The resultant erosion risk map is shown in Figure 8.33. The SLEMSA model takes into account the erosive power of rain, interception by plant cover, resistance of the soil to erosion and several topographic measures. The exact nature of the model to be used for this study is still to be defined, but it is hoped that the integration of the spatial database and remotely sensed data will provide more accurate predictions for the planning authorities.

Within the information system several queries can be generated such as: find all areas with a land use other than forest on slopes greater than 18%. This is a useful by-product of the database generation providing a surveillance system. The product of this particular query identifies areas of illegal activity in the Philippines.

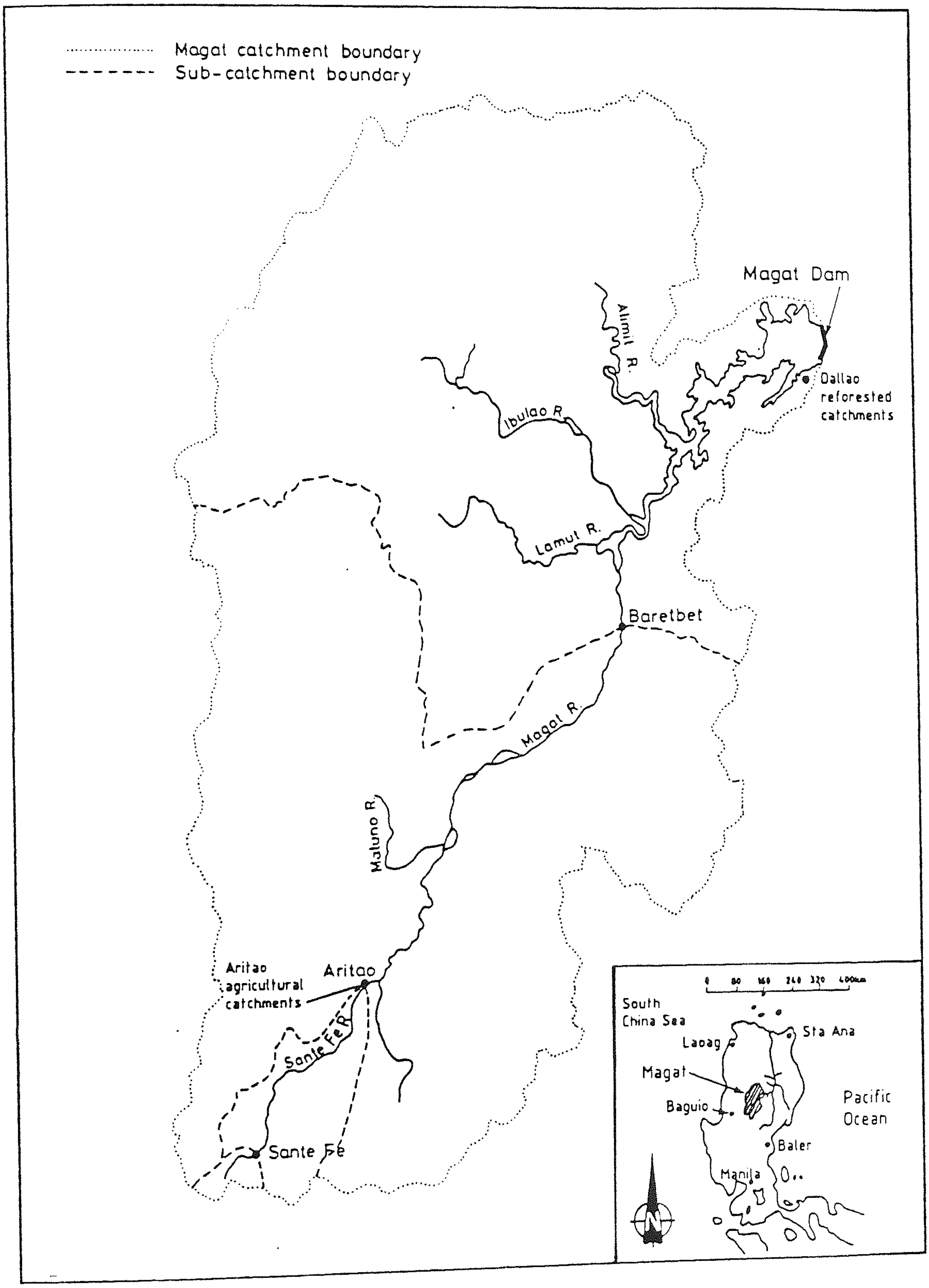


Figure 8.22. The Magat River Catchment.

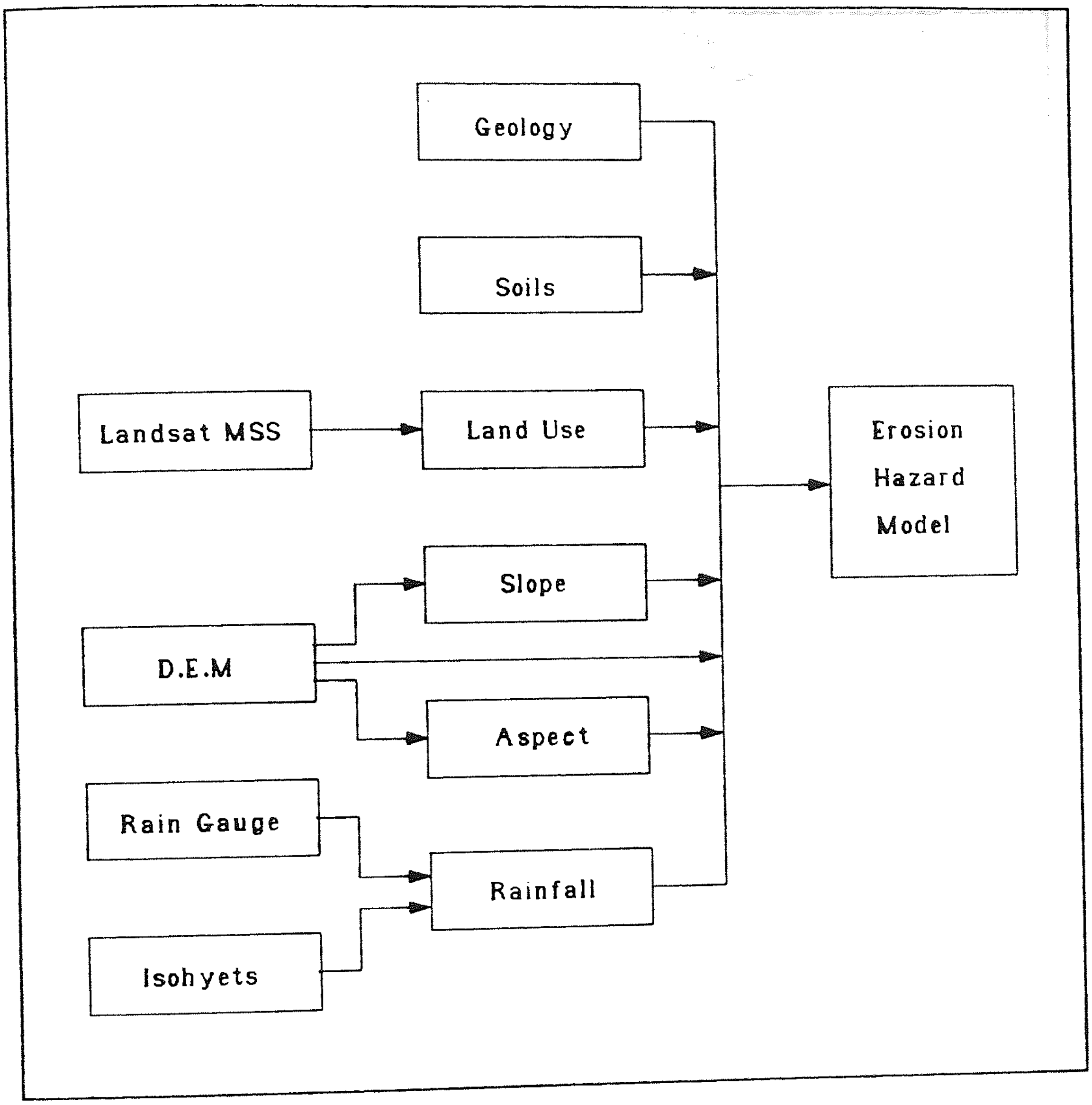


Figure 8.23. Magat Database Components.

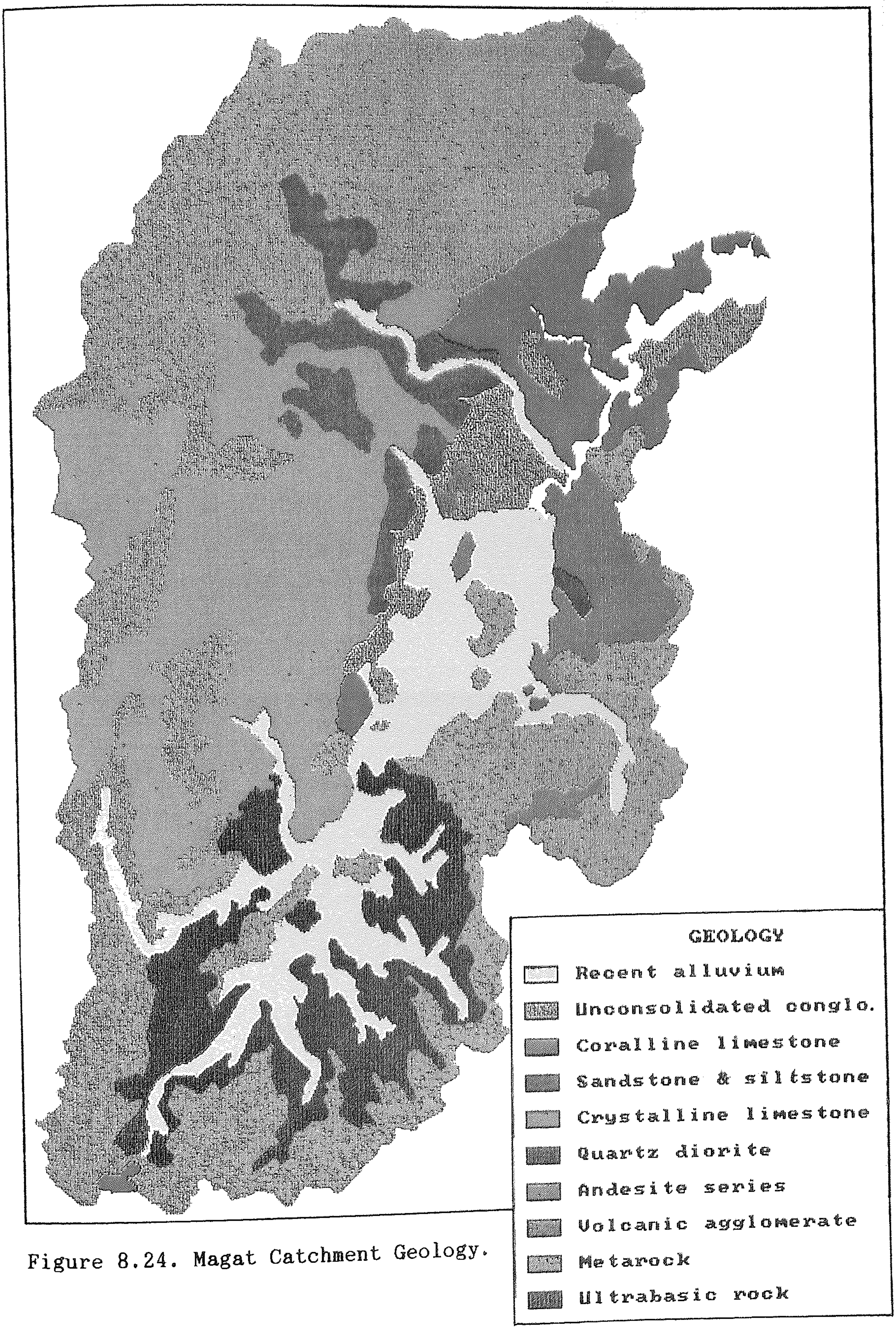


Figure 8.24. Magat Catchment Geology.

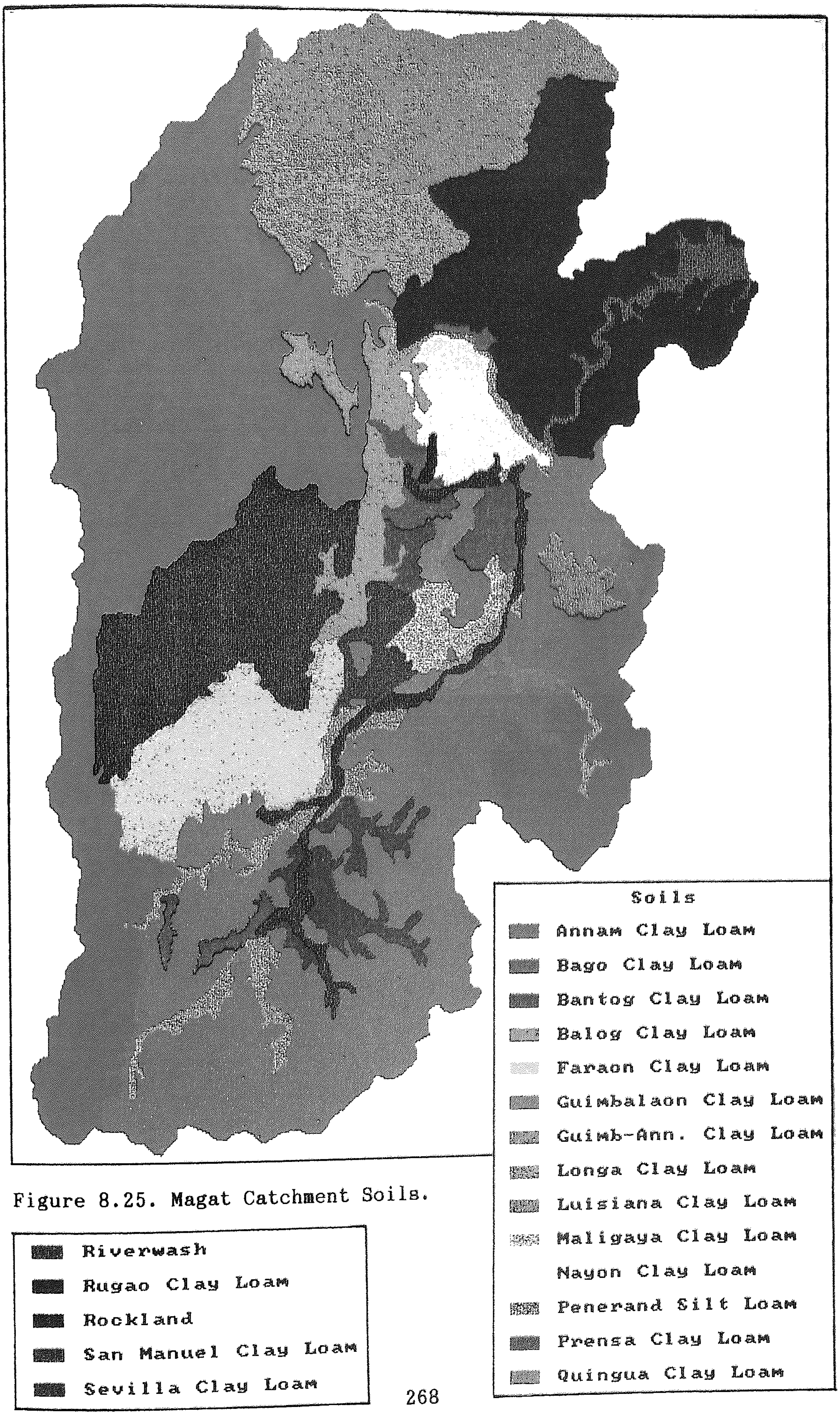


Figure 8.25. Magat Catchment Soils.

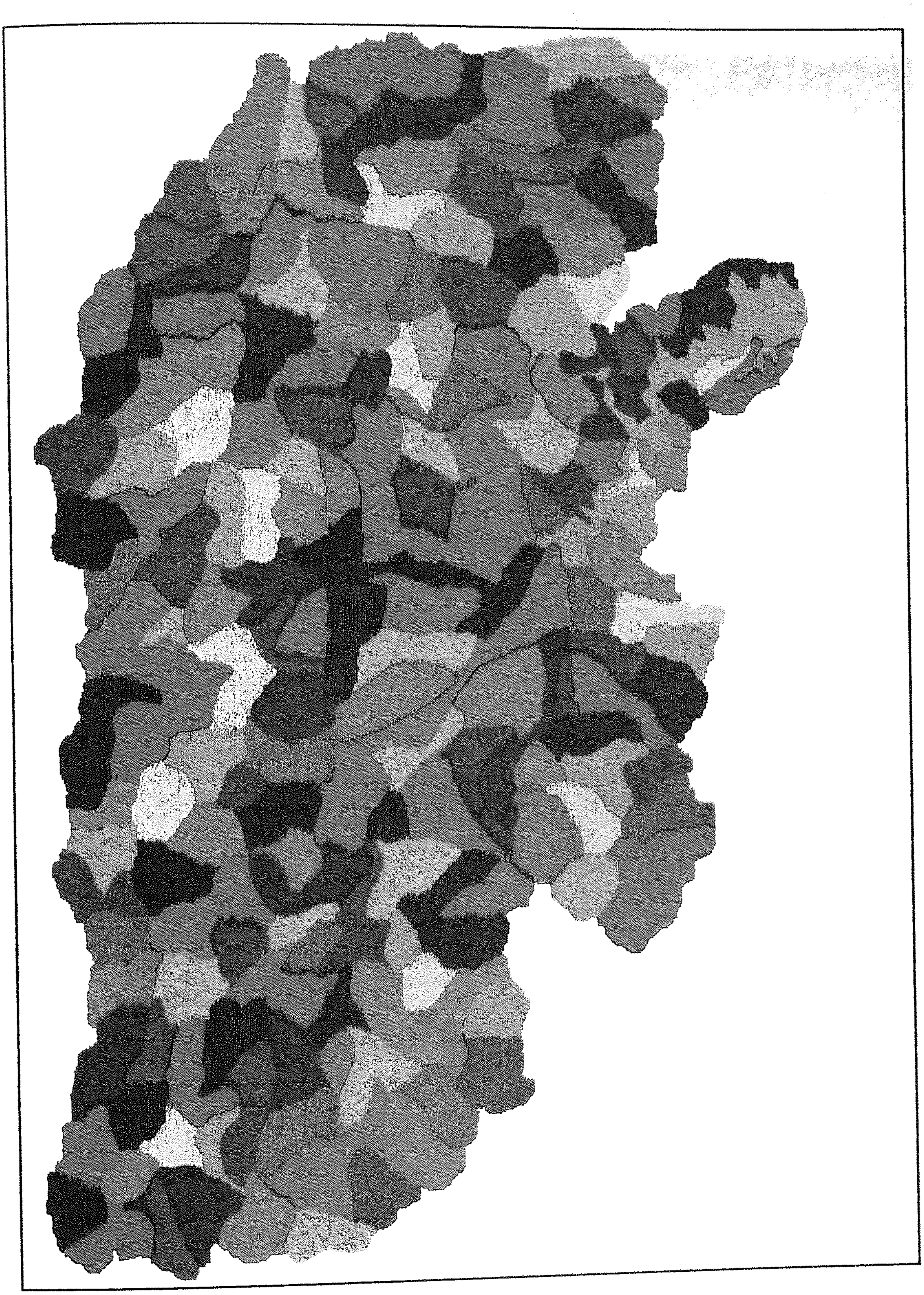


Figure 8.26. Magat Catchment Sub-Catchments.

Figure 8.27. Landsat MSS Magat Catchment.

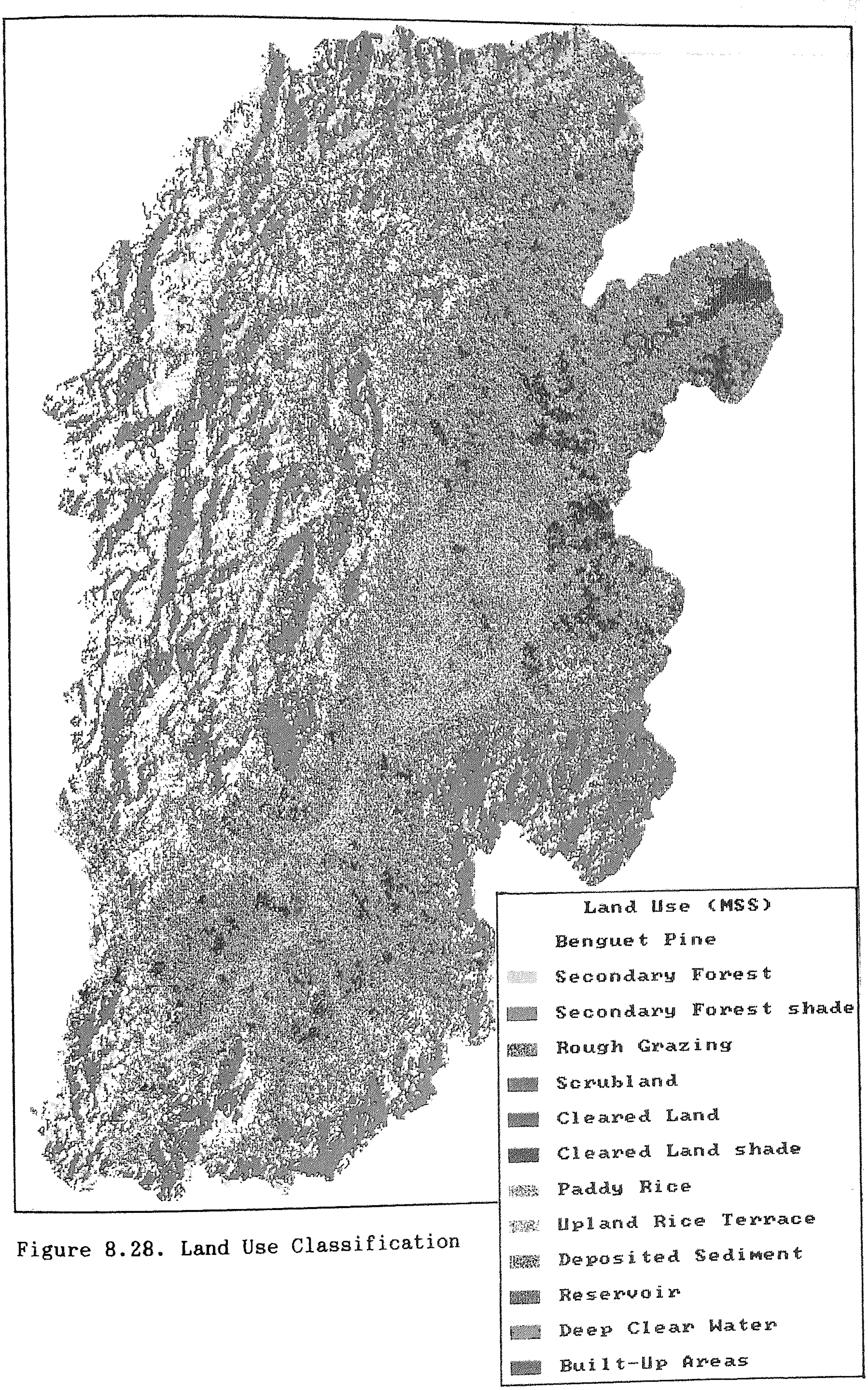


Figure 8.28. Land Use Classification

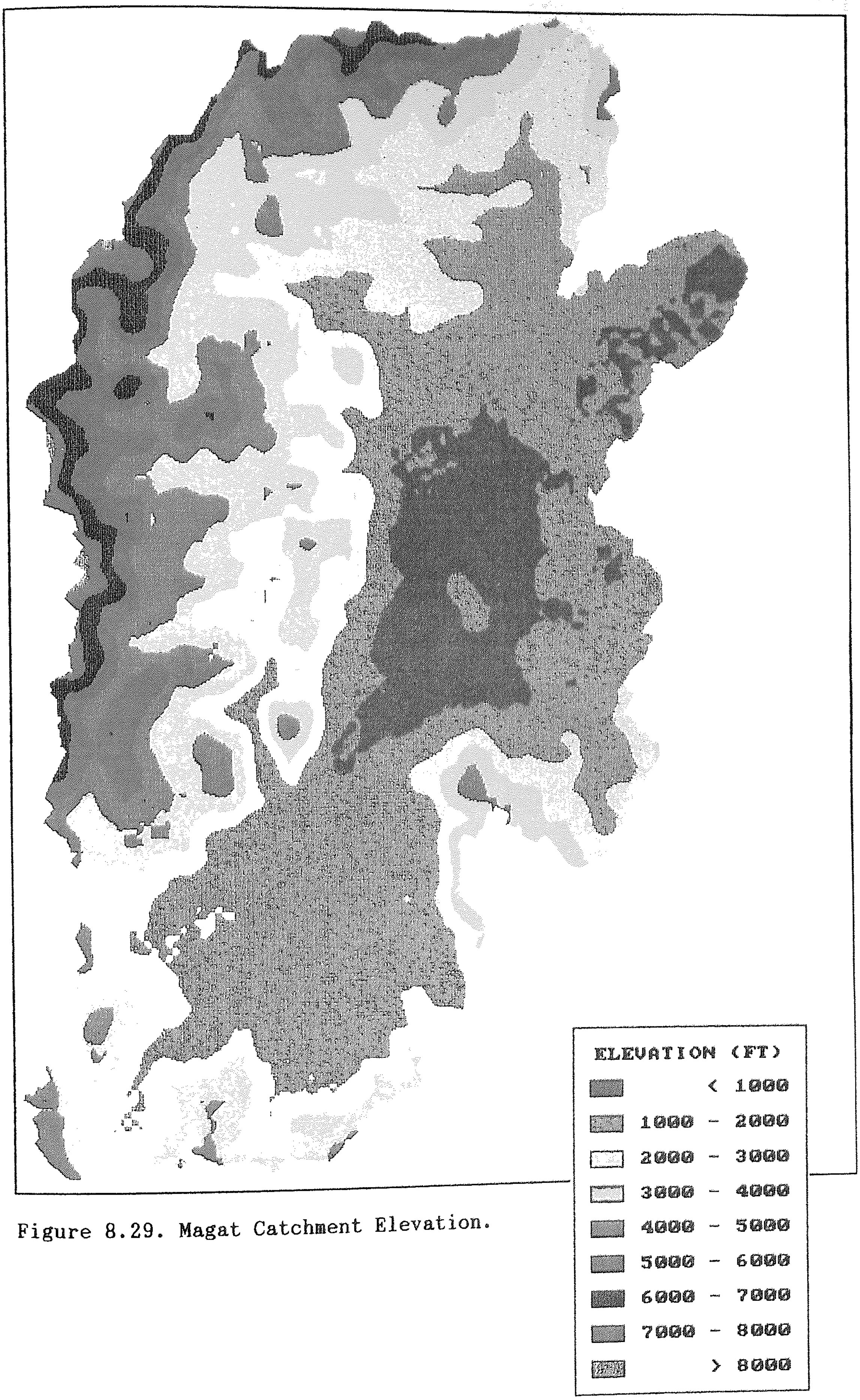


Figure 8.29. Magat Catchment Elevation.

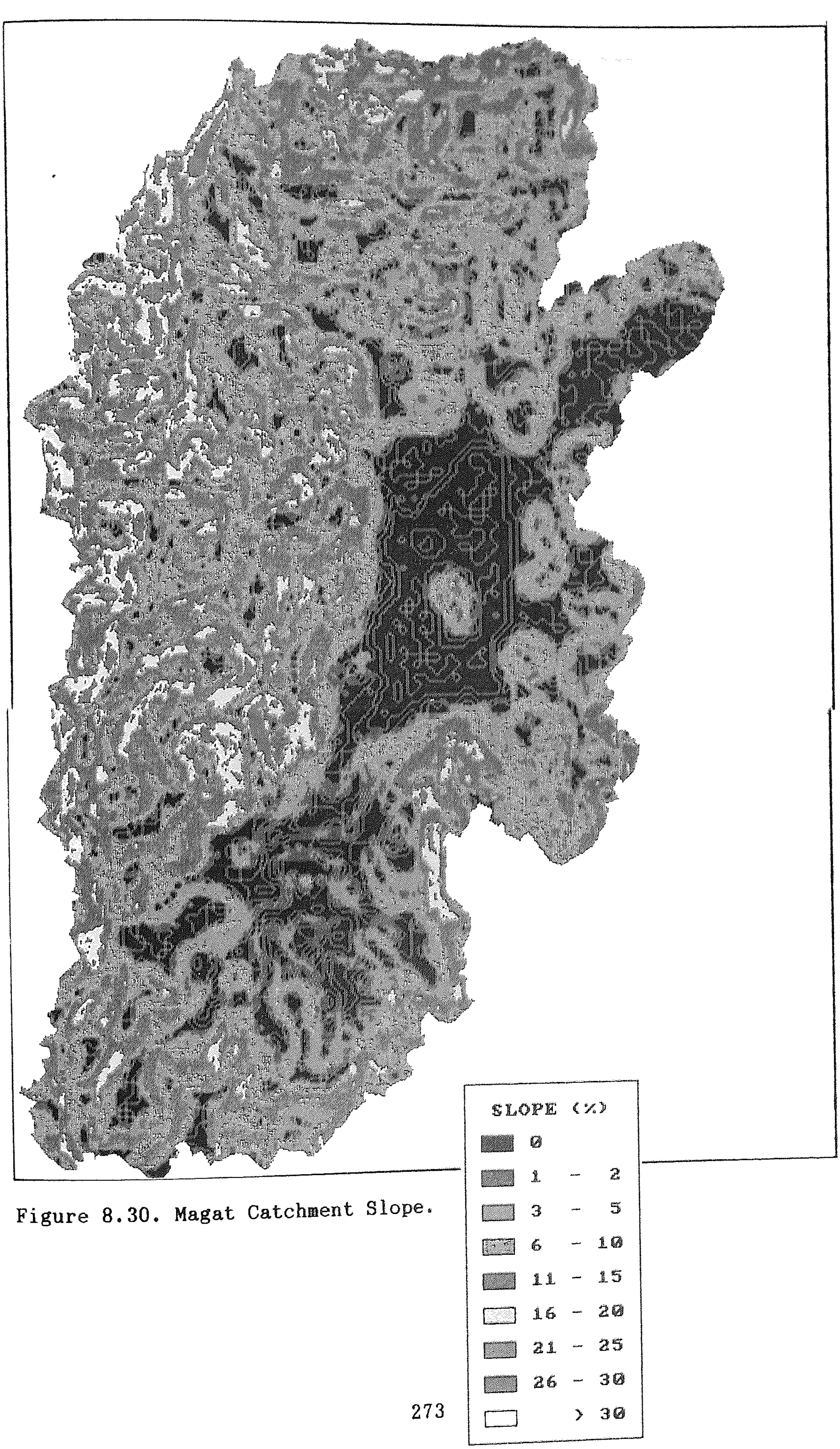


Figure 8.30. Magat Catchment Slope.

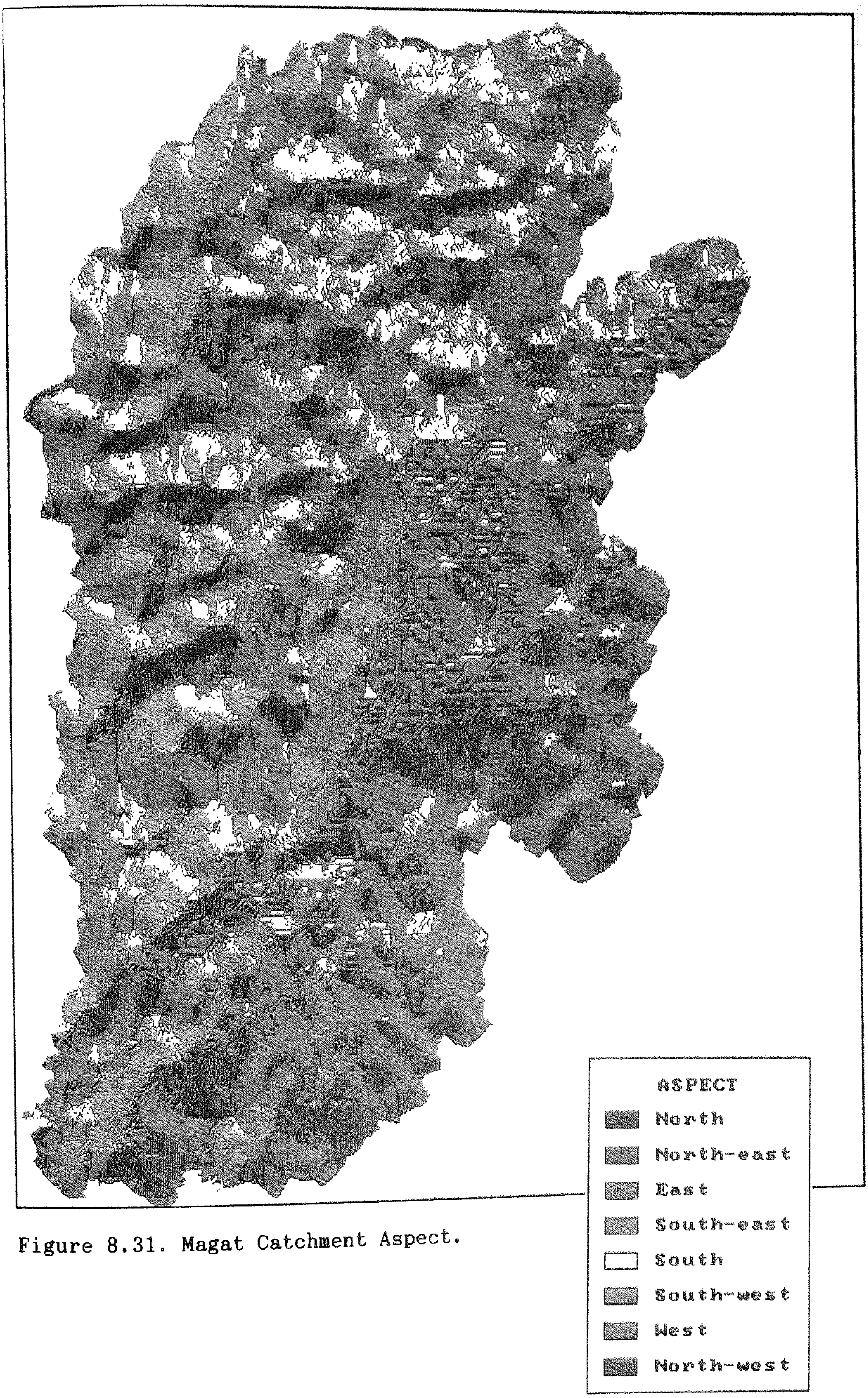


Figure 8.31. Magat Catchment Aspect.

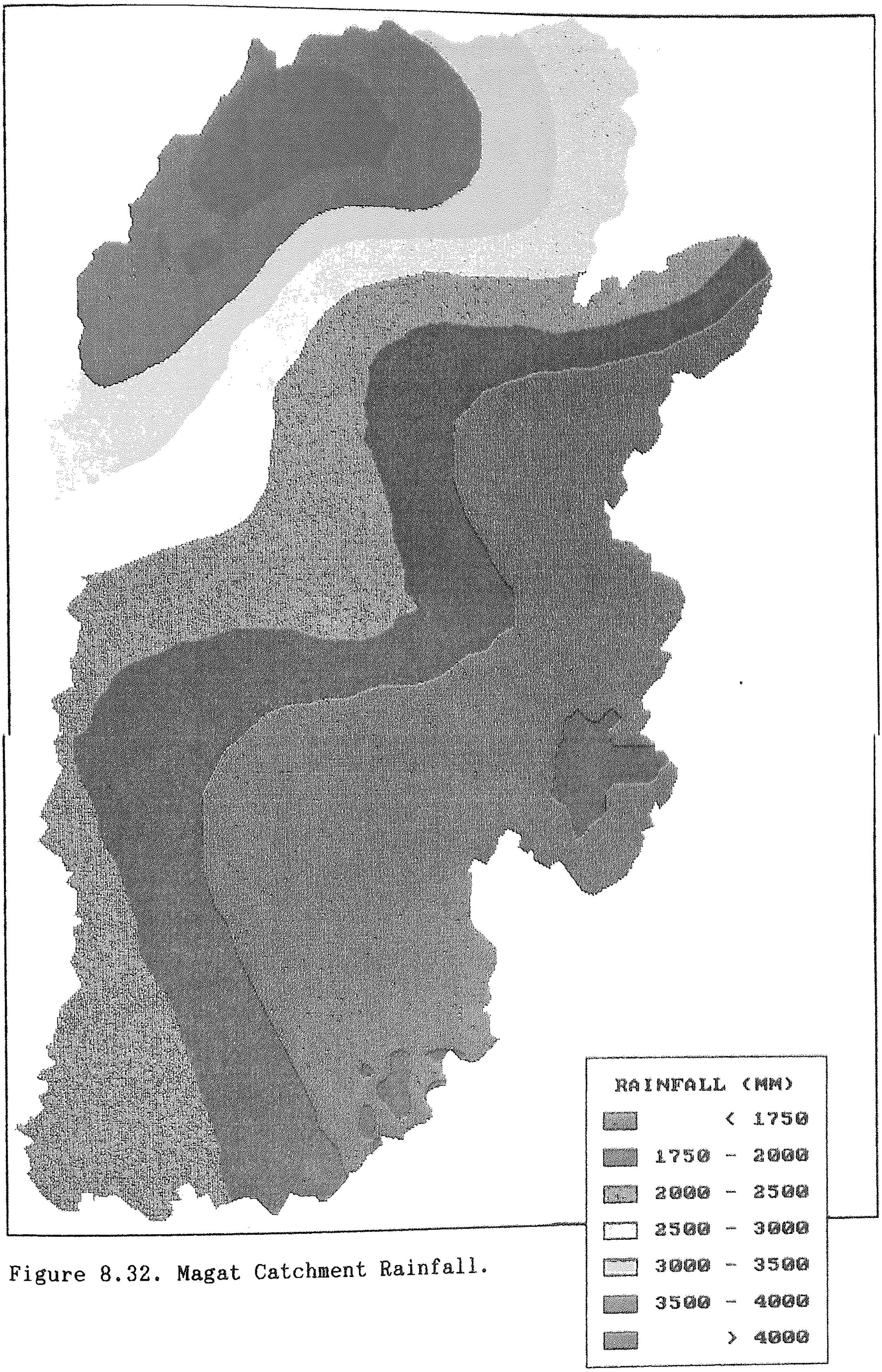


Figure 8.32. Magat Catchment Rainfall.

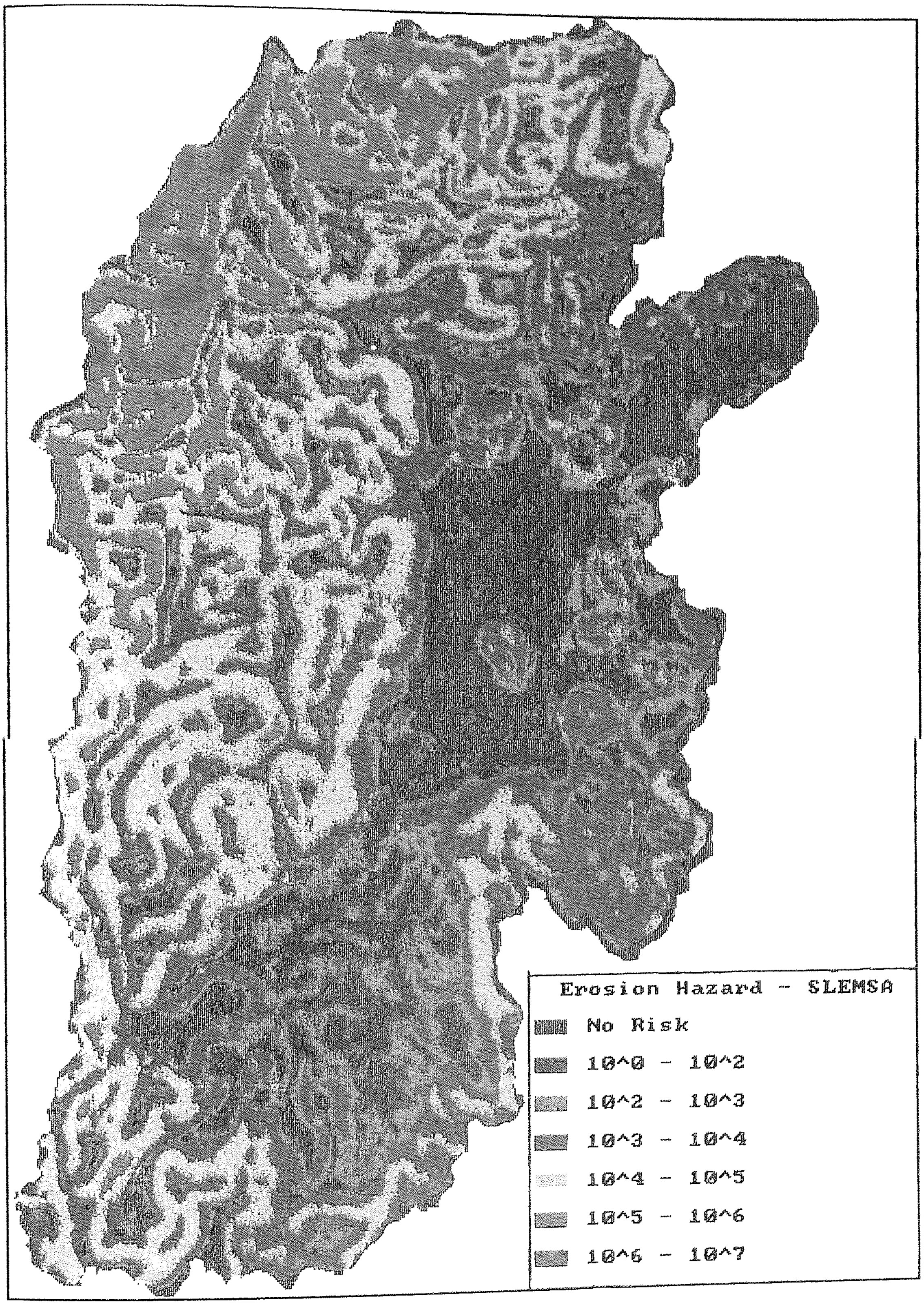


Figure 8.33. Magat Catchment Erosion Hazard.

9 Conclusions

Information, and a means to access information, are becoming an increasingly important aspect of the work of the Civil Engineer. Many operational design and project management applications require access to new information and frequently need to update existing information. This thesis looks at some of the problems a River Basin manager may encounter, and suggests a number of practical ways of providing sufficient information to solve these problems.

The elements of originality explored in this thesis are in:

- data acquisition
- data storage and retrieval
- data integration
- manipulation of spatially related data
- implementation of man-machine interfaces

Due consideration is given to financial aspects, the emphasis being on low cost solutions, affordable in the context of project economics, and therefore being applicable to lesser developed countries.

The basic tool is an integrated water information management system - a combination of micro-computer technology, data communications networks and sensors. The use of micro-computers is becoming increasingly commonplace throughout the world. This is attributable to the decreased cost and increased availability of semi-conductor technology. Many lesser developed countries are already producing their own micro-computers. This makes the use of micro-computer based technology increasingly attractive to foreign governments with limited hard currency.

The aim of the water information management system is to assist the water resources manager in the acquisition, analysis and prediction of spatial and temporal variation of hydrological data. The system therefore assists in the derivation of river basin management strategies, and is also a means of monitoring the effect of adopted policies.

Data acquisition is an important aspect of any information system. In a project life cycle the acquisition of data can occupy as much as 95% of the initial available resources. Within the project, data acquisition can have three main sources: cartographic data, telemetered point source data and remotely sensed imagery. The acquisition of data remains under the control, and within the budget, of the project management.

The vast quantities of cartographic data available are slowly being converted to digital format. Currently this activity is mainly restricted to the developed world. Consequently the first task of many applications is the production of a map database, which is generally the most time consuming, and hence the most costly, phase of a project. For any innovative method to replace data entry by bit-pad digitiser, it must demonstrate its cost effectiveness. This involves higher accuracy, faster and more economic operation, usability and limited training requirements. In most projects data entry is performed with minimum supervision by the lowest grade of operator and therefore more complex methods of data entry, requiring more operator skill, prove impractical unless substantial time or cost savings are obtained. Automatic and semi-automatic methods of data extraction (line following algorithms) are consequently not considered as they do not yet meet these requirements.

The method proposed in this study resembles the use of bit-pad digitisers, in that features are manually traced using a cursor, but the method proposed is easier to use, and results in a reduction in digitisation errors. Perhaps the most significant disadvantage of using a bit pad digitiser is that the digitised features are not marked in any way. Using video/raster digitisation to capture an image of the map enables an exact record of the digitised features to be presented on the display. If portability is a concern, video cameras are significantly easier to transport than digitising tablets. However this method requires each operator to digitise features on an image display, and will therefore not be a viable alternative to bit pad digitisation until the total cost of micro-computer, image processor, video camera and display becomes cheaper than a digitising tablet.

The digitisation process generates a topographic and thematic database for subsequent applications. Once completed the database can be archived, requiring only minor updates, and forms the basis for all data integration on the project. The themes of the database relate to the nature of the project, but for water resources management possible useful themes include: geology, soils, elevation, slope, aspect, land use, transportation networks and drainage networks.

The digital elevation model (DEM) is one of the most useful databases, and possible the most time consuming to produce. Commercially available DEMs have either limited resolution or limited coverage, and in many cases the only alternative is to independently produce the DEM within the system. This involves digitising of contours from topographic maps and interpolation from the ordered random contour vertex points to a regular fine grid. Once generated the DEM can be used to produce a number of useful derived products; including slope, aspect, watershed partitioning/sub-catchments, inter-visibility, topographic shading and simulated oblique viewing. The methods explored in this thesis make use of the hardware features of the technology, using framestores for storage of spatial components of data, image processing techniques and on-board processors to manipulate the data, which therefore reduce the processor and memory requirements of the host computer.

Data collection platforms provide a means of collecting hydrological data from remote locations and transmitting recorded data triggered by a clock on a regular time interval or by an event occurrence. Data can be transmitted to the information management system via a telephone communications network, or alternatively using the DCP transmission communications channels on board the geostationary (Meteosat, GOES, GMS), or polar orbiting (NOAA-TIROS) meteorological satellites.

Remotely sensed imagery from geostationary and polar orbiting meteorological satellites can be acquired by a number of methods. For the acquisition to remain under the control of the project management the only possible alternatives are reception directly from the satellites or transmission via direct line to ground receiving stations.

A number of the database themes can also be derived from large or medium scale remotely sensed imagery. For example land use, transport and drainage networks can be either automatically or manually derived from airborne or low altitude satellite remote sensing. The use of low altitude satellite remote sensing, for example Landsat and SPOT, can be prohibitively expensive. The main application would be in the initial phases of a project typically for determining land use or spatial variation of water quality. The cost of acquisition and length of procurement time for these products precludes their use in real-time and short term monitoring systems. However, for long term monitoring projects, use of these systems can prove cost-effective.

The same reasoning could be applied to the aerial photographic survey. Unsurpassed for large scale topographic and photogrammetric survey, the aerial photograph could most sensibly be used in the initial phases of the project. This thesis has detailed an innovative, low cost alternative to the airborne photographic survey i.e. airborne videography. Videography can be seen as a method of acquiring data, under control of the project management, at a spatial resolution between that of satellite imagery and aerial photography.

At present (1989) airborne video cannot provide the geometric fidelity of the aerial photograph for photogrammetric mapping, it can however provide an adequate basis for topographic mapping and land use change detection. The advantages offered by videography over photography are;

the ease with which video data can be integrated into spatial information systems; the potential for multi-spectral processing on standard image processing systems; and the immediate realisation of the success of a survey through simultaneous recording and viewing, and immediate playback. Airborne video has distinct improved possibilities for the future, especially with the introduction of digital video tape recorders and the new 1024 line television standards.

Spatial data, by its very nature, requires large volumes of storage. The exact storage requirements differ according to the hardware dependent features of the system. For example a vector based system stores an area as a series of connected vertices, a raster based system stores an area as a set of contiguous connected values. The nature of these structures has been studied as part of this thesis and two basic structures have been developed for storage and retrieval of vector and raster data. Various data compression techniques were also investigated for storage of raster data.

The value of any individual data source can be significantly increased if integrated with other data. This thesis describes the use of a spatial information system for providing the necessary utilities for integration, storage, retrieval and manipulation of data from as widely divergent sources as data collection platforms (point source monitoring), remote sensing (spatial monitoring), maps and survey charts, data archives and statistical analysis.

It is generally not sufficient to provide only techniques for the acquisition, storage and retrieval of data. In addition to these functions, the river basin manager must be able to use these data intelligently, to extract new data from existing data, and export data to other processes such as hydrological or hydraulic models. For useful data to be extracted, the information stored in the system and its functionality must be clearly presented, in a easily accessible format - the intelligent man-machine interface.

In the past computer scientists have concentrated on the production of increasingly powerful processors and larger volumes of memory. This has resulted in large numbers of computer systems requiring highly skilled operators. Over the last few years the cost of producing the software components of computer systems has risen drastically, in some cases as high as 90% of the total system cost. The necessity for efficient programming methodologies and software engineering is therefore apparent.

With the decrease in cost and increased availability of computer systems the design of the man machine interface (MMI) has increased in prominence. The assessment of usability of engineering computer systems is now a major contributing factor in computer sales to all but academic institutions. The necessity for a graphics and menu based interface can be demonstrated by the success of 'WIMP' (Windows.Icons.Menus.Pointers) interfaces on micro-computers such as the Apple Macintosh.

The user interface and spatial information systems technology therefore forms a major part of the water information system developed as part of this research. The system makes use of a graphical, mouse and menu driven interface, to provide pathways through the complexities of spatial data acquisition, processing and analysis tools. A particular problem can often be solved by a number of possible routes within the system.

Query of spatially related data stored in the system, using relational database constructs, enables the intelligent retrieval of data. Spatial query, in terms of geographical location, is relatively straightforward and consists of graphical clipping operations. The incorporation of topology is more complicated. The majority of applications involving topology search for information relative (in space) to a selected feature. For example, to find all areas suitable for growing irrigated sugar within 10km of a major road. This can be achieved by producing a thematic map showing all areas within 10km of a major road - a proximity corridor. The proximity corridor is then overlayed on the result of the area attribute query. This query highlights the inadequacies of standard relational databases, and query languages such as SQL. This example also introduces the concept of overlaying multiple class attributes for a given area, to derive further themes - water availibility, land suitability, from existing themes; and suggests the use of an Artificial Intelligence approach in developing a rule base from expert knowledge.

The production of the total database including permanent themes such as geology, elevation and soils; semi-permanent themes such as land use, and transient themes such as rainfall and soil moisture, provides the basis for distributed modelling of a number of hydrological processes such as interception, erosion, runoff and channel flow. Calibration of these models over a period of time makes use of telemetered point source data. The end product is a system that assists the water resources manager in making more accurate predictions, and increases the efficiency of water management policies through improved simulation of hydrological processes.

The applications of the water information system explored in this thesis have shown part of the potential of the system in the field of thematic database production. These applications have included generation of input data for hydrological and hydraulic models; in drainage design; in the mapping of land use related to groundwater pollution; and in the modelling of erosion risk and sediment yield in a reservoir catchment.

Above all this thesis has demonstrated that using low cost technology it is possible to build information systems for a wide variety of applications in River Basin Management. It is believed that no-one to date has demonstrated, to this degree, the economic integration of cartographic data, telemetered point source data, satellite imagery and airborne video survey.

BIBLIOGRAPHY

Ackermann, F., 1978, **"Experimental Investigation into the Accuracy of Contouring from DTM"**, *Photogrammetric Engineering and Remote Sensing: Special DTM Issue*, Vol. 44, No. 12, pp. 1537-1548.

Alföldi, T.T., J.C.Munday, 1978, **"Water Quality Analysis by Digital Chromaticity Mapping of Landsat Data"**, *Canadian Journal of Remote Sensing*, Vol. 4, No. 2, pp. 108-126.

Allam, M.M., 1978, **"DTM Application in Topographic Mapping"**, *Photogrammetric Engineering and Remote Sensing: Special DTM Issue*, Vol. 44, No. 12, pp. 1513-1520.

Allewijn, R., 1988, **"Regional Hydrological Systems Analysis using Satellite Remote Sensing data and a Geographical Information System"**, *International Journal of Remote Sensing: Special Issue - 1987 EARSeL Symposium*, Vol. 9, Nos. 10 & 11, pp. 1775-1785.

Amos, C.L., T.T.Alföldi, 1979, **"The Determination of Suspended Sediment Concentration in a Macrotidal System using Landsat Data"**, *Journal of Sedimentary Petrology*, Vol. 49, No. 1, pp. 159-174.

Anderson, D.G., 1979, **"Satellite Versus Conventional Methods in Hydrology"**, *Satellite Hydrology*, pp. 33-36, American Water Resources Association.

Aranuvachapun, S., 1985, **"Satellite Remote Sensing of Ocean Colours"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 513-521.

Arneson, M.A., 1981, **"Blending Transportation Facilities into the Landscape - Computer Style"**, *Proceedings American Society of Photogrammetry Workshop for Automated Photogrammetry and Cartography*, pp. 97-105.

Aronoff, S., G.A.Ross, 1984, **"Detection of Environmental Disturbance using Colour Aerial Photography and Thermal Infrared Imagery"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 48, No. 4, pp. 587-591.

Arthur, D.W.G., 1973, **"Interpolation of a Function of Many Variables, II"**, *Photogrammetric Engineering*, Vol. 39, No. 2, pp. 261-266.

Ayeni, O.O., 1982, **"Optimum Sampling for Digital Terrain Models: A Trend Towards Automation"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 48, No. 11, pp. 1687-1694.

Band, L.E., 1985, **"Digital Elevation Models and Hydrologic Information Systems"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 201-207.

Band, L.E., 1986, **"Topographic Partition of Watersheds with Digital Elevation Models"**, *Water Resources Research*, Vol. 22, No. 1, pp. 15-24.

Bartolucci, L.A., B.F.Robinson, L.F.Silva, 1977, **"Field Measurement of the Spectral Response of Natural Waters"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 43, No. 5, pp. 595-598.

Batson, R.M., K.Edwards, E.M.Eliason, 1975, **"Computer-Generated Shaded Relief Images"**, *Journal of Research of the United States Geological Survey*, Vol. 3, No. 4, pp. 401-408.

Bauer, F.L., 1971, **"Software Engineering"**, *Information Processing 71*, Amsterdam: North Holland Publishing Co.

Baussart, M., 1968, **"Photogrammetry and Civil Engineering in France"**, *The Photogrammetric Record*, 6(31), pp. 7-15.

Beasley, D.B., 1977, **"ANSWERS: A Mathematical Model for Simulating the Effects of Land Use and Management on Water Quality"**, PhD Thesis, Purdue University, West Lafayette, Indiana, USA.

Beaumont, T.E., J.J.Burke, J.Davie, M.J.Leese, S.P.R.Vincent, 1988, **"GIST: A Geographic Information System Toolkit for Water Resources and Engineering Applications"**, *Scott Wilson Kirkpatrick/Wimpey Laboratories Marketing Literature.*

Bhargava, D.S., 1983, **"Very Low Altitude Remote Sensing of the Water Quality of Rivers"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 49. No. 6, pp. 805-809.

Blanchard, B.J., R.W.Leamer, 1973, **"Spectral Reflectance of Water Containing Suspended Sediment"**, *Proceedings of American Water Resources Association Symposium on Remote Sensing and Water Resources Management*, pp. 339-347.

Blyth, K., 1983, **"Remote Sensing and Water Resources Engineering"**, *Proceedings of Joint Royal Aerospace Establishment and Transport and Road Research Laboratory Workshop*, Vol. 2, pp. H1-H62.

Blyth, K., G.P.Nash, 1978, **"Aerial Infrared Photography for Flood Plain Investigations"**, *Proceedings of the Institute of Hydrology*, pp. 425-434.

Boyle, A.R., 1979, **"Scan Digitisation of Cartographic Data"**, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 27-46, Academic Press.

Bukata, R.P., J.E.Bruton, J.H.Jerome, 1983, **"Use of Chromaticity in Remote Measurement of Water Quality"**, *Remote Sensing of Environment*, Vol. 13, pp. 161-177.

Burroughs, P.A., 1986, *Principles of Geographic Information Systems for Land Resources Assessment*, Oxford University Press, N.Y.

Calkins, H.W., 1984, **"A Pragmatic Approach to Geographic Information Systems Design"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Calkins, H.W., 1984, **"Creating Large Digital Files from Mapped Data"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Cameron, E.A., 1984, **"Manual Digitising Systems"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Carpenter, D.J., 1981, **"Inland Water Quality Assessment from Landsat Data"**, *Proceedings of the 15th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp. 483-490.

Carpenter, D.J., S.M.Carpenter, 1983, **"Modeling Inland Water Quality using Landsat Data"**, *Remote Sensing of Environment*, Vol. 13, pp. 345-352.

Carstensen, L.W.(Jr.), 1986, **"Regional Land Information Systems Development using Relational Databases and Geographic Information Systems"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 507-516.

Carter, J.R., 1988, **"Digital Representations of Topographic Surfaces"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1577-1584.

Catlow, D.R., 1986, **"The Multi-Disciplinary Applications of DEMs"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 447-454.

Chen, Q., 1986, **"The Design of a Spatial Database Mangement System"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 423-432.

Cheshire, H.M., S.Khorram, J.A.Brockhaus, 1985, **"Monitoring Estuarine Water Quality from Landsat TM"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 227-236.

Clark, D.K., J.B.Zaitzeff, L.V.Strees, W.S.Glidden, 1974, **"Computer Derived Coastal Water Classifications via Spectral Signatures"**, *Proceedings of the 9th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 1213-1239.

Collins, M., C.Pattiaratchi, 1984, **"Identification of Suspended Sediment in Coastal Waters using Airborne Thematic Mapper Data"**, *International Journal of Remote Sensing*, Vol. 5 No. 4, pp. 635-657.

Collins, S.H., G.C.Moon, 1981, **"Algorithms for Dense Digital Terrain Models"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 47, No. 1, pp. 71-76.

Colwell, R.N.,ed., 1983, *Manual of Remote Sensing*, 2nd ed. American Society of Photogrammetry and Remote Sensing.

Cowen, D.J., 1988, **"GIS versus CAD versus DBMS: What are the Differences?"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1551-1556.

Cracknell, A.P. ed., 1981, *Remote Sensing in Meteorology, Oceanography and Hydrology*, Ellis Horwood.

Cracknell, A.P., N.McFarlane, K.McMillan, J.A.Charlton, 1982, **"Remote Sensing in Scotland using Data Received from Satellites: A Study of the Tay Estuary Region using Landsat MSS Imagery"**, *International Journal of Remote Sensing*, Vol. 3, No. 2, pp. 113-137.

Dangermond, J., 1984, **"A Classification of Software Components commonly used in Geographic Information Systems"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Dangermond, J. C.Friedman, 1984, **"ARC/INFO : A Modern Geographic Information System"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Dave, J.V., R.Bernstein, 1982, **"Effect of Terrain Orientation and Solar Position on Satellite-Level Luminance Observations"**, *Remote Sensing of Environment*, Vol. 12, pp. 331-348.

Davis, J.C., M.J.McCullagh, eds., 1975, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, Wiley.

Dealy, B.T., 1985, **"A Raster Based Data Entry System"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 433-437.

Deimal, L.E.(Jr.), R.J.Fornaro, D.F.McAllister, C.L.Doss, 1982, **"Techniques for Computerised Lake and River Fills in Digital Terrain Models"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 48, No. 9, pp. 1431-1436.

Delfiner, P., J.P.Delhomme, 1975, **"Optimum Interpolation by Kriging"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 96-114, Wiley.

Dettmann, T.R., 1988, *DOS Programmers Reference*, Que.

Deutsch, M., D.R.Wiesnet, A.Rango, eds., 1981, *Satellite Hydrology*, American Water Resources Association.

Devereux, B.J., 1986, **"The Integration of Cartographic Data Stored in Raster and Vector Formats"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 257-266.

Doyle, F.J., 1978, **"Digital Terrain Models: An Overview"**, *Photogrammetric Engineering and Remote Sensing: Special DTM Issue*, Vol. 44, No. 12, pp. 1481-1485.

Dubayer, R.O., J.Dozier, 1986, **"Orthographic Terrain Views Using Data Derived from Digital Elevation Models"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 52, No. 4, pp. 509-518.

Ebner, H., P.Reiss, 1984, **"Experience with Height Interpolation by Finite Elements"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 50, No. 2, pp. 177-182.

Ellehoj, E., G.D.Lodwick, 1983, **"Potential Applications of Kriging in Remote Sensing"**, *Australian Computer Bulletin*, Sept. 1973, pp. 12-15.

Engman, E.T., 1984, **"Remote Sensing Based Continuous Hydrologic Modelling"**, *Advanced Space Research*, Vol. 4, No. 11, pp. 201-209.

Environmental Systems Research Institute, 1988, *ARC/INFO Marketing Literature*.

Everitt, J.H., P.R.Nixon, 1985, **"False Colour Video Imagery: A Potential Remote Sensing Tool for Range Management"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 51, No. 6, pp. 675-679.

Fisher, P.F., G.G.Wilkinson, 1985, **"Towards the Knowledge Based Information System"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 51-55.

Fleming, G., 1969, **"Design Curves for Suspended Load Estimation"**, *Proceedings of the Institution of Civil Engineers*, 43, pp. 1-9.

Fleming, G., R.Walker, 1976, **"A Runoff Erosion Model for Land Use Assessment and Management"**, Technical Report, Department of Civil Engineering, University of Strathclyde.

Foley, J.D., A.Van Dam, 1984, *Fundamentals of Interactive Computer Graphics*, 2nd Ed. Addison-Wesley.

Fournier, F., 1960, **"Climat et erosion, la retention entre l'erosion du sol par l'eau et les precipitations atmospheriques"**, Presses Universitaires de France, Paris.

Frank, A.U., 1988, **"Requirements for a Database Management System for a GIS"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1557-1564.

Freeman, H., G.G.Pieroni eds., 1979, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, Academic Press.

Freeman, H., P.P.Loutrel, 1962, **"An Algorithm for the Solution of the Two Dimensional 'Hidden-Line' Problem"**, *IEEE Transactions on Electronic Computing*, Vol. EC 16, No. 6, pp. 784-790.

Goran, W.D., M.Shapiro, J.D.Westervelt, 1986, **"GRASS Imagery - Grid Interface"**, *Proceedings of Symposium of Commission IV of the International Society for Photogrammetry and Remote Sensing: Mapping from Modern Imagery*, pp. 430-437.

Grist, M.W., 1972, **"Digital Ground Models: An Account of Recent Research"**, *The Photogrammetric Record*, 7(40), pp. 424-441.

Häme, T., M.Rantasuo, 1988, **"Shuttered Camera - Aerial Colour Video Imaging in the Visible and Near Infrared"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 12, pp. 1735-1738.

Hannah, M.J., 1981, **"Error Detection and Correction in Digital Terrain Models"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 47, No. 1, pp. 63-69.

Haralick, R.M., 1979, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 63-100, Academic Press.

Hardy, R.L., 1971, **"Multiquadric Equations of Topography and Other Irregular Surfaces"**, *Journal of Geophysical Research*, Vol. 76, No. 8, pp. 1905-1915.

Hardy, R.L., 1977, **"Least Squares Prediction"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 43, No. 4, pp. 475-492.

Herschy, R.W., E.C.Barrett, J.N.Roozekrans, 1988, **"The World's Water Resources - A Major Neglect"**, *European Space Agency Report ESA-BR-40*.

HMSO, 1987, *Handling Geographic Information: Report of the Committee of Enquiry chaired by Lord Chorley*, HMSO.

Imaging Technology Incorporated, 1987, *PCVisionplus Frame Grabber User's Manual*.

Jancaitis, J.R, J.L.Junkins, 1973, **"Modelling Irregular Surfaces"**, *Photogrammetric Engineering*, Vol. 39, No. 4, pp. 413-420.

Jensen, R.W., C.C.Tonies, 1979, *Software Engineering*, Prentice-Hall.

Jenson, S.K., J.O.Domingue, 1988, **"Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1593-1600.

Johnson, R.W., 1978, **"Mapping of Chlorophyll-A Distributions in Coastal Waters"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 44, No. 5, pp. 617-624.

Johnson, R.W., R.C.Harriss, 1980, **"Remote Sensing for Water Quality and Biological Measurement in Coastal Waters"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 46, No. 1, pp. 77-85.

Jones, C.B., 1985, **"Hierarchical Data Structures for Scale-Independent Geographic Information Systems"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 57-66.

Junkin, B.G., 1982, **"Development of Three Dimensional Spatial Displays using a Geographic Based Information System"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 48, No. 4, pp. 577-586.

Junkins, J.L., J.R.Jancaitis, 1974, **"Piecewise Continuous Surfaces: A Flexible Basis for Digital Mapping"**, *Proceedings of the American Society of Photogrammetry Fall Convention*, pp. 1-18.

Junkins, J.L., J.R.Jancaitis, G.W.Miller, 1972, **"Smooth Irregular Curves"**, *Photogrammetric Engineering*, Vol. 38, No. 6, pp. 565-573.

Justice, C.O., S.W.Wharton, B.N.Holben, 1981, **"Application of Digital Terrain Data to Quantify and Reduce the Topographic Effect on Landsat Data"**, *International Journal of Remote Sensing*, Vol. 2, No. 3, pp. 213-229.

Katibah, G.H., 1981, **"Map Graphics and the Real World"**, *Proceedings American Society of Photogrammetry Workshop for Automated Photogrammetry and Cartography*, pp. 105-113.

Keisteri, T., 1986, **"FINGIS - Software and Data Manipulation"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 69-75.

Khorram, S., 1981, **"Use of Ocean Colour Scanner Data in Water Quality Mapping"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 47, No. 5, pp. 667-676.

Khorram, S., 1981, **"Water Quality Mapping from Landsat Digital Data"**, *International Journal of Remote Sensing*, Vol. 2, No. 2, pp. 145-153.

Khorram, S., 1985, **"Development of Water Quality Models Applicable Throughout the Entire San Francisco Bay and Delta"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 51, No. 1, pp. 53-62.

Khorram, S., H.M.Cheshire, 1985, **"Remote Sensing of Water Quality in the Neuse River Estuary, North Carolina"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 51, No. 3, pp. 329-341.

Kingslake, R., 1986, *An Introductory Course in Computer Graphics*, Chartwell-Bratt.

Kleiner, A., K.E.Brassel, 1986, **"Hierarchical Grid Structures for Static Geographic Databases"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 485-496.

Klemas, V., A.M.F.Abdel-Kader, 1982, **"Remote Sensing of Coastal Processes with Emphasis on the Nile Delta"**, *Proceedings of the E.R.I.M International Symposium Remote Sensing of the Environment - 1st Thematic Conference*, pp. 389-415.

Klemas, V., D.Bartlett, W.Philpot, R.Rogers, L.Reed, 1974, **"Coastal and Estuarine Studies with ERTS-1 and Skylab"**, *Remote Sensing of Environment*, Vol. 3, pp. 153-174.

Klemas, V., J.F.Borchardt, W.M.Treasure, 1973, **"Suspended Sediment Observations from ERTS-1"**, *Remote Sensing of Environment*, Vol. 2, pp. 205-221.

Klemas, V., M.Otley, W.Philpot, C.Wethe, R.Rogers, N.Shah, 1974, **"Correlation of Coastal Water Turbidity and Circulation with ERTS-1 and Skylab Imagery"**, *Proceedings of the 9th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 1289-1317.

Klooster, S.A., J.P.Scherz, 1974, **"Water Quality by Photographic Analysis"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 40, No. 8, pp. 927-935.

Knapp, E.L., 1979, **"Spatial Data Integration"**, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 47-62, Academic Press.

Knuth, D.E., 1979, *The Art of Computer Programming: Volume 1 - Fundamental Algorithms*, Addison-Wesley.

Kramber, W.J., A.J.Richardson, P.R.Nixon, K.Lulla, 1988, **"Principal Components Analysis of Aerial Video Imagery"**, *International Journal of Remote Sensing*, Vol. 9, No. 9, pp. 1415-1422.

Kraus, K., E.M.Mikhail, 1972, **"Linear Least-Squares Interpolation"**, *Proceedings of 12th Congress of the International Society of Photogrammetry*, pp. 1016-1029.

Kritikos, H., L.Yorinks, H.Smith, 1974, **"Suspended Sediments Analysis using ERTS-A Data"**, *Remote Sensing of Environment*, Vol. 3, pp. 69-78.

Kuo, C.Y., 1978, **"Laboratory Requirements for In-Situ and Remote Sensing of Suspended Material: Final Report"**, *NASA Technical Report NASA-CR-145367*, 90 pages.

Lathrop, R.G.(Jr.), T.M.Lillesand, 1986, **"Use of Thematic Mapper Data to Assess Water Quality in Green Bay and Cebtral Lake Michigan"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 52, No. 5, pp. 671-680.

Lillesand, T.M., R.W.Kiefer, 1979, *Remote Sensing and Image Interpretation*, Wiley.

Lillesand, T.M., F.L.Scarpace, J.L.Clapp, 1975, **"Water Quality in Mixing Zones"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 41, No. 3, pp. 285-298.

Lin, C.C., A.J.Chen, D.C.Chern, 1986, **"New Approach to Semi-Automatically Generate Digital Elevation Data by Using a Vidicon Camera"**, *Proceedings of the* 7^{th} *Symposium Commission VII of the International Society for Photogrammetry and Remote Sensing: Remote Sensing for Resources Development and Environmental Management*, pp. 430-437.

Lindell, L.T., 1981, **"Mapping of Water Quality using Landsat Imagery"**, *Proceedings of the* 15^{th} *E.R.I.M International Symposium on Remote Sensing of the Environment*, pp. 1375-1385.

Lindell, L.T., O.Steinvall, M.Jonsson, T.H.Claesson, 1985, **"Mapping of Coastal Water Turbidity using Landsat Imagery"**, *International Journal of Remote Sensing*, Vol. 6, No. 5, pp. 629-642.

Lintz, J., D.S.Simonett, eds., 1976, *Remote Sensing of Environment*, Addison-Wesley.

Lo, C.P., 1976, **"Photographic Analysis of Water Quality Changes"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 42, No. 3, pp. 309-315.

Loutrel, P.P., 1970, **"A Solution to the Hidden-Line Problem for Computer Drawn Polyhedra"**, *IEEE Transactions on Computers*, Vol. C 19, No. 3, pp. 205-213.

Maggio, R.C., R.D.Baker, 1988, **"Techniques for Acquisition of Aerial Video Imagery"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 2, pp 229-231.

Marble, D.F., 1983, **"Geographic Information Systems and Remote Sensing"**, *Manual Of Remote Sensing*, 2nd ed. American Society of Photogrammetry and Remote Sensing.

Marble, D.F., H.W.Calkins, D.J.Peuquet, eds., 1984, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Mark, D.M., 1986, **"The Use of Quadtrees in Geographic Information Systems and Spatial Data Handling"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 1, pp. 517-526.

Maul, G.A., H.R.Gordon, 1975, **"On the Use of the Earth Resource Technology Satellite (Landsat-1) in Optical Oceanography"**, *Remote Sensing of Environment*, Vol. 4, pp. 95-128.

Maul, G.A., R.L.Charnell, R.H.Qualset, 1974, **"Computer Enhancement of ERTS-1 Images for Ocean Radiances"**, *Remote Sensing of Environment*, Vol. 3, pp. 237-253.

Mauser, W., 1984, **"Calculation of Flood Hydrographs using Landsat-Derived Land-Use Information in the Dreisam Watershed, South-West Germany"**, *Advanced Space Research*, Vol. 4, No. 11, pp. 211-216.

McKim, H.L., C.J.Merry, R.W.Layman, 1984, **"Water Quality Monitoring using an Airborne Spectroradiometer"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 50, No. 3, pp. 353-356.

McManus, J., 1984, **"Gradients of Change in the Estuarine Environments of the Tay"**, *Proceedings of the Remote Sensing Society Workshop on Remote Sensing Applications in Civil Engineering, ESA Publication - ESA-SP-216*, pp. 143-151.

Meijerink, A.M.J., C.M.Valuenzuela, A.Stewart, 1988, *The Integrated Land and Watershed Management Information System*, ITC Publication No. 7.

Meisner, D.E., 1985, **"Fundamentals of Airborne Video Remote Sensing"**, *Remote Sensing of Environment*, Vol. 19, pp. 63-79.

Meisner, D.E., 1985, **"Colour Infrared Aerial Video: Applications in Natural Resource Management"**, *Proceedings of 10th William T.Pecora Memorial Remote Sensing Symposium*, pp. 80-87.

Meisner, D.E., O.M.Lindstrom, 1985, **"Design and Operation of a Colour Infrared Aerial Video System"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 51, No. 5, pp. 555-560.

Miller, C.L., 1957, **"The Spatial Model Concept of Photogrammetry"**, *Photogrammetric Engineering*, Vol. 23, No. 1, pp. 31-35.

Miller, C.L., R.A.Laflamme, 1958, **"The Digital Terrain Model - Theory and Application"**, *Photogrammetric Engineering*, Vol. 24, No. 3, pp. 433-442.

Munday, J.C.(Jr.), P.L.Zubkoff, 1981, **"Remote Sensing of Dinoflagellate Blooms in a Turbid Estuary"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 47, No. 4, pp. 523-531.

Munday, J.C., 1983, **"Chromaticity of Path Radiance and Atmospheric Correction of Landsat Data"**, *Remote Sensing of Environment*, Vol. 13, pp. 525-538.

Munday, J.C., T.T.Alföldi, 1975, **"Chromaticity Changes from Isoluminous Techniques used to Enhance Multispectral Remote Sensing Data"**, *Remote Sensing of Environment*, Vol. 4, pp. 221-236.

Munday, J.C., T.T.Alföldi, 1979, **"Landsat Test of Diffuse Reflectance Models for Aquatic Suspended Solids Measurement"**, *Remote Sensing of Environment*, Vol. 8, pp. 169-183.

Muralikrishna, I.V., 1984, **"Optical Remote Sensing for Coastal Zone Management"**, *Proceedings of the Remote Sensing Society Workshop on Remote Sensing Applications in Civil Engineering, ESA Publication - ESA-SP-216*, pp. 175-193.

Natural Environmental Research Council, 1988, *Geographic Information in the Environmental Sciences: Report of the Working Group on Geographic Information*, NERC.

Newmann, W.M., R.F.Sproull, 1979, *Principles of Interactive Computer Graphics*, McGraw-Hill.

Nixon, P.R., D.E.Escobar, R.L.Bowen, A.J.Richardson, 1984, **"Video Colour Infrared Imagery: A Future Natural Resource Managament Tool"**, *Proceedings IX Biennial Workshop on Colour Aerial Photography in the Plant Sciences*, pp. 159-165.

Nixon, P.R., D.E.Escobar, R.M.Menges, 1985, **"A Multiband Video System for Quick Assessment of Vegetal Condition and Discrimination of Plant Species"**, *Remote Sensing of Environment*, Vol. 17, pp. 203-208.

Palmer, R.A., R.C.Maggio, P.T.Sprinz, 1987, **"Digital Analysis of Colour Infrared Aerial Video Imagery"**, *Proceedings of 11th Colour Photography and Videography Workshop of the American Society of Photogrammetry*, pp. 280-285.

Parker, D.H., 1988, **"The Unique Qualities of a Geographic Information System: A Commentary"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1547-1550.

Peucker, T.K., M.Tichenor, W.D.Rase, 1975, **"The Computer Version of Three Relief Representations"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 187-197.

Peuquet, D.J., A.R.Boyle, 1984, **"Interactions between the Cartographic Document and the Digitising Process"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Pfaltz, J.L., 1975, **"Representation of Geographic Surfaces within a Computer"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 210-230, Wiley.

Press, W.H., B.P.Flannery, S.A.Teukolsky, W.T.Vetterling, 1988, *Numerical Recipes in C - the Art of Scientific Computing*, Cambridge University Press.

Ragan, R.M., J.D.Fellows, 1979, **"Computer-aided Watershed Analysis using Remote Sensing Based Regional Information Systems"**, *The Contribution of Space Observations to Water Resources Management*, Pergamon Press.

Rais, J., 1986, **"The Development of an Integrated Geographic Information System for Indonesia"**, *Proceedings of Symposium of Commission IV of the International Society for Photogrammetry and Remote Sensing: Mapping from Modern Imagery*, pp. 84-84.

Real World Graphics, 1988, *PC4000 Graphics Board Users Guide*.

Riazonoff, S., B.Cervelle, J.Chorowicz, 1988, **"Ridge and Valley Line Extraction from Digital Terrain Models"**, *International Journal of Remote Sensing*, Vol. 9, No. 6, pp. 1175-1183.

Ritchie, J.C., F.R.Schiebe, J.R.McHenry, 1976, **"Remote Sensing of Suspended Sediment in Surface Waters"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 42, No. 12, pp. 1539-1545.

Ritter, P., 1987, **"A Vector-Based Slope and Aspect Generation Algorithm"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 8, pp. 1109-1111.

Salomonson, V.V., P.D.Bavsar, eds., 1979, *The Contribution of Space Observations to Water Resources Management*, Pergammon Press.

Scarpace, F.L., B.W.Quirk, R.W.Kiefer, S.L.Wynn, 1981, **"Wetland Mapping from Digitised Aerial Photography"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 47, No. 7, pp. 829-838.

Schachter, B., 1979, **"Computer Generation of Shaded Relief Maps"**, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 355-368, Academic Press.

Scherz, J.P., D.R.Graff, W.C.Boyle, 1969, **"Photographic Charcateristics of Water Pollution"**, *Photogrammetric Engineering*, Vol. 35, No. 1, pp. 38-43.

Schmidt, A.H., W.A.Zafft, 1975, **"Programs of the Harvard University Laboratory for Computer Graphics and Spatial Analysis"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 231-243, Wiley.

Schultz, G.A., 1988, **"Remote Sensing in Hydrology"**, *Journal of Hydrology*, Vol. 100, pp. 239-265.

Schut, G.H., 1974, **"Two Interpolation Methods"**, *Photogrammetric Engineering*, Vol. 40, No. 12, pp. 1447-1453.

Science and Engineering Research Council, 1989, *A New Initiative in the Applications of Information Technology to the Construction and Transport Industries*, Environment Committee circular, Feb. 1989.

Shapiro, L.G., 1979, **"Design of a Spatial Information System"**, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 101-118, Academic Press.

Sharpnack, D.A., G.Akin, 1969, **"An Algorithm for Computing Slope and Aspect from Elevations"**, *Photogrammetric Engineering*, Vol. 35, No. 3, pp. 247-248.

Smith, S.E., K.H.Mancy, A.F.A.Latif, 1982, **"Monitoring Water Quality and Environmental Changes in the Aswan High Dam Reservoir from Landsat Imagery"**, *Proceedings of the E.R.I.M International Symposium Remote Sensing of the Environment - 1st Thematic Conference*, pp. 375-388.

Smith, S.E., K.H.Mancy, A.F.A.Latif, E.A.Fosnight, 1982, **"Assessment of Sedimentation in the Aswan Reservoir using Landsat Imagery"**, *Proceedings of the E.R.I.M International Symposium Remote Sensing of the Environment - 1st Thematic Conference*, pp. 727-737.

Smith, T.R., D.J.Peuquet, 1985, **"Control of Spatial Search for Complex Queries in a Knowledge Based Geographic Information System"**, *Proceedings of the International Conference of the Remote Sensing Society and the Centre for Earth Resources Management - Advanced Technology for Monitoring and Processing Global Environmental Data*, pp. 439-454.

Sprunt, B.F., 1975, **"Hidden Line Removal from Three-Dimensional Maps and Diagrams"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 198-209, Wiley.

Sprunt, B.F., 1975, **"Relief Representation in Automated Cartography: An Algorithmic Approach"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 173-186, Wiley.

Stocking, M., 1987, **"A Methodology for Erosion Hazard Mapping of the SADCC Region"**, *SADCC Co-ordination Unit*, Report No.9, Maseru, Lesotho.

Stortz, K., M.Sydor, 1974, **"Remote Sensing of Western Lake Superior"**, *Proceedings of the 9th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 933-937.

Swain, P.H., S.M.Davis, eds., 1978, *Remote Sensing: The Quantitative Approach*, McGraw-Hill

Tanchotikul, A., 1987, **"Investigation of Physical Condition of Water Masses using NOAA Satellite AVHRR Data in the Gulf of Thailand"**, *Proceedings Remote Sensing '87: Remote Sensing and Information Systems for Planning and Development*, TS.V.C.3.

Tassan, S., 1981, **"A method for the Retrieval of Phytoplankton and Suspended Sediment Concentrations from Remote Measurement of Water Colour"**, *Proceedings of the 15th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 577-586.

Thomas, I.L., 1980, **"Suspended Sediment Dynamics from Repetitive Landsat Data"**, *International Journal of Remote Sensing*, Vol. 1, No. 3, pp. 285-292.

Thorpe, L.W., 1986, **"Graphical Display and Manipulation in Two and Three Dimensions of Digital Cartographic Data"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 2, pp. 418-430.

Tomlinson, R.F., A.R.Boyle, 1984, **"The State of Development of Systems for Handling Natural Resource Inventory Data"**, *Basic Readings in Geographic Information Systems*, SPAD Systems, Ltd.

Toomey, P., 1985, **"A Microlight Video Imaging System for Low Level Remote Sensing"**, MSc Thesis, University College, London.

TYDAC Technologies Ltd., 1988, *SPANS Marketing Literature*.

Usery, E.L., P.Altheide, R.R.P.Deister, D.J.Barr, 1988, **"Knowledge-Based GIS Techniques Applied to Geological Engineering"**, *Photogrammetric Engineering and Remote Sensing: Special GIS Issue*, Vol. 54, No. 11, pp. 1623-1628.

Valenzuela, C.R., 1988, **"ILWIS Overview"**, *ITC Journal: Special ILWIS Issue*, No. 1, pp. 4-14.

Verdin, J.P., 1985, **"Monitoring Water Quality Conditions in a Large Western Reservoir with Landsat Imagery"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 51, No. 3, pp. 343-353.

Vlcek, J., 1983, **"Videography: Some Remote Sensing Applications"**, *Proceedings of the 1983 American Society of Photogrammetry Symposium*, pp. 63-69.

Ward, R.C., 1975, *Principles of Hydrology*, McGraw-Hill.

Werner, H., 1979, **"Numerical Algorithms for Interpolation and Smoothing"**, *Proceedings of NATO Advanced Study Institute on Map Data Processing*, pp. 331-354, Academic Press.

Whitley, S.L., R.W.Pearson, B.R.Seyfarth, M.H.Graham, 1981, **"ELAS: A Geobased Information System that is Transferrable to Several Computers"**, *Proceedings of the 15th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 1091-1098.

Whitlock, C.H., 1978, **"Laboratory and Field Measurement of Upwelled Radiance and Reflected Spectra of Suspended James River Sediments near Hopewell, Virginia"**, *NASA Technical Report NASA-TP-1292*, 30 pages.

Whitlock, C.H., C.Y.Kuo, S.R.LeCroy, 1982, **"Criteria for the Use of Regression Analysis for Remote Sensing of Sediment and Pollutants"**, *Remote Sensing of Environment*, Vol. 12, pp. 151-168.

Whitlock, C.H., W.G.Witte, J.W.Usry, E.A.Gurganus, 1978, **"Penetration Depth at Green Wavelengths in Turbid Waters"**, *Photogrammetric Engineering and Remote Sensing*, Vol. 44, No. 11, pp. 1405-1410.

Wirth, N., 1974, **"On the Composition of Well-Structured Programs"**, *Computing Surveys*, Vol. 6, No. 4, pp. 247-259.

Wischmeier, W.H., D.D.Smith, 1978, **"Predicting Rainfall Erosion Losses: A Guide to Conservation Planning"**, *USDA-SEA Agr. Handbook 537*, Agricultural Research Service, USDA, Washington.

Witte, W.G., 1981, **"Laboratory Measurement of Physical, Chemical and Optical Characteristics of Lake Chicot Sediment Waters"**, *NASA Technical Report NASA-TP-1941*, 30 pages.

Wong, F., R.Orth, D.E.Friedman, 1981, **"The Use of DTM in the Rectification of Satellite-Borne Imagery"**, *Proceedings of the 15th E.R.I.M International Symposium on Remote Sensing of the Environment*, pp 653-662.

Wood, T.R., 1987, **"Remote Sensing Needs in the 1990's for Hydrology/Water Management Practice in the UK"**, *Paper presented at EARSeL Symposium - Looking Back/Looking Forward European Remote Sensing Needs in the 1990s.*

Woodsford, P.A., 1986, **"Cartographic Digitising - Technical Trends and Economic Factors"**, *Proceedings Auto Carto - International Conference on the Acquisition, Management and Presentation of Spatial Data*, Vol. 2, pp. 489-495.

Yoeli, P., 1975, **"Compilation of Data for Computer-Assisted Relief Cartography"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 352-368.

Yoeli, P., 1975, **"Methodology of Computer Assisted Cartography"**, *NATO Advanced Study Institute on Display and Analysis of Spatial Data*, pp. 130-136, Wiley.

Young, W.H., 1981, **"Digital Terrain Mapping and Things It Can Do"**, *Proceedings American Society of Photogrammetry Workshop for Automated Photogrammetry and Cartography*, pp. 23-43.

Young, W.H., D.M.Isbell, 1978, **"Production Mapping with Orthophoto Digital Terrain Models"**, *Photogrammetric Engineering and Remote Sensing: Special DTM Issue*, Vol. 44, No. 12, pp. 1521-1536.

APPENDIX A

Source Listings

Introduction

This appendix details a number of routines developed as part of the water information system. The application of these routines is in the graphical user interface to the system and in compression of thematic map data in the spatial information system. As such these routines are deeply embedded within the system, and demonstrate the utility of structured modular programming in developing flexible routines. In the top-down/bottom up programming methodologies these routines make use of graphics and I/O primitives in the form of 'C' library functions, and are therefore the 'bottom' of the user interface routines.

The routines are written in 'C', compiled using the Microsoft C Compiler; and linked to the C large model, Essential Graphics, MSMouse and RWG-PC4000 subroutine libraries.

The functionality of the routines is described in Chapter 7.

A.1 Code Fragment 1 - The Locator

```
/* Define cursor active window */

#define MINX -128
#define MAXX +640

#define MINY  -32
#define MAXY +736

mcursor(px,py,cxx,cxy,cyx,cyy,col,m)
int *px,*py,cxx,cxy,cyx,cyy,m[];
unsigned char col;

/* Routine to generate cross hair cursor on image display */
/* and write position on EGA screen. */

{
      int i,j,mx,my,nx,ny,bl,bm,br,f;
      int x,y,z,tx,ty,panx,pany;
      char *cx,*cy,bufx[10],bufy[10]
      unsigned char rc,gc,bc,rcol,gcol,bcol;
      unsigned char rcur[15];
      unsigned char gcur[15];
      unsigned char bcur[15];

      /* Initialise board and component variables */

      pcinit();
      sorg(128,32);
      pansc(1,384,288);

      /* Initialise Mouse */

      m_instal();

      /* Initialise absolute position and zoom counters */

      x=tx=*px;
      y=ty=*py;
      z=1;

      /* Determine cursor colour from colour and LUTs */

      defcol(col,&rcol,&gcol,&bcol);

      /* Read under cursor in all components*/
      /* Returns colour of pixel at centre of cross */

      rc=rcursor(x,y,1,rcur);
      gc=rcursor(x,y,2,gcur);
      bc=rcursor(x,y,3,bcur);

      /* plot cross hair cursor on all components */

      pcursor(x,y,1,rcol,rc);
      pcursor(x,y,2,gcol,gc);
      pcursor(x,y,3,bcol,bc);

      /* Write cursor position to EGA screen */

      cx=itoa(x,bufx,10);
      fhatsay(0,cx,m[7],cxx,cxy);
      cy=itoa(y,bufy,10);
      fhatsay(0,cy,m[7],cyx,cyy);

      /* Start loop to move cursor - initialise motion counters */

      m_motion(&mx,&my);
      do{
            /* Determine mouse motion since last call */

            m_motion(&mx,&my);

            /* Increment position counters and validate */

            x=x+mx;
            y=y-my;
```

```
if(x<MINX)
      x=MINX;
else if(x>MAXX-1)
      x=MAXX-1;
if(y<MINY)
      y=MINY;
else if(y>MAXY-1)
      y=MAXY-1;

/* If current position differs from last */

if(tx!=x || ty!=y){

      /* Erase position plotted on EGA screen */

      grrtulc(cxx,cxy-8,24,8,m[2]);
      grrtulc(cyx,cyy-8,24,8,m[2]);

      /* Restore image over old cursor */

      wcursor(tx,ty,1,rcur);
      wcursor(tx,ty,2,gcur);
      wcursor(tx,ty,3,bcur);

      /* Read under cursor at new position */

      rc=rcursor(x,y,1,rcur);
      gc=rcursor(x,y,2,gcur);
      bc=rcursor(x,y,3,bcur);

      /* plot cross hair cursor */

      pcursor(x,y,1,rcol,rc);
      pcursor(x,y,2,gcol,gc);
      pcursor(x,y,3,bcol,bc);

      /* Write cursor position to EGA screen */

      cx=itoa(x,bufx,10);
      fhatsay(0,cx,m[7],cxx,cxy);
      cy=itoa(y,bufy,10);
      fhatsay(0,cy,m[7],cyx,cyy);
      tx=x;
      ty=y;
}

/* If left button pressed increment zoom counter */

m_press(0,&bl,&mx,&my);

if(bl>0)
      z=z*2;

/* If right button pressed decrement zoom counter */

m_press(1,&br,&mx,&my);

if(br>0)
      z=z/2;

/* Clip zoom to valid values */

if(z<1)
      z=1;
else if(z>16)
      z=16;

/* Change image zoom and pan - */

azoom(1,z,z);

/* Pan image to cursor */

if(z>1){
      panx=x+128;
      pany=y+32;
}
else{
      if(x<5)
            panx=384-(5-x);
      else if(x>507)
            panx=384+(x-507);
```

```
                        if(y<5)
                                pany=288-(5-y);
                        else if(y>507)
                                pany=288+(y-507);
                }
                pansc(1,panx,pany);

                /* If middle button pressed exit loop */

                m_press(2,&bm,&mx,&my);
        }
        while(bm==0);

        /* Restore image over cursor */

        wcursor(tx,ty,1,rcur);
        wcursor(tx,ty,2,gcur);
        wcursor(tx,ty,3,bcur);

        /* Return x and y values of point defined */

        *px=x;
        *py=y;

        sorg(0,0);
}
```

A.2 Code Fragment 2 - The Valuator

```
selint(cx,cy,min,max,def,m,bg)
int cx,cy,min,max,def,m[],bg;

/* Valuator routine using mouse pad as range of possible values */
/* scaled to fit min-max range, set to default value def and */
/* written to EGA display at cx,cy */

{
        int mx,my,nx,ny,bl,br;
        int val,dx;
        char *cval,bufv[10];
        float scf;

        /* Install mouse */

        m_instal();

        /* Determine scaling on 0-639, min-max */

        scf=639.0/(max-min);

        /* Set default mouse position as def */

        dx=(def-min)*scf;
        m_setcur(dx,100);

        /* Enter valuator loop */

        do{
                /* Determine mouse position and button status */

                m_posbut(&mx,&my,&bl,&br);

                /* Evaluate position to value and check validity */

                val=(mx/scf)+min;

                if(val>max)
                        val=max;
                else if(val<min)
                  val=min;

                /* Write position to EGA display */

                cval=itoa(val,bufv,10);

                if(val<0)
                        fhatsay(0,cval,m[5],cx-8,cy);
                else
                        fhatsay(0,cval,m[5],cx,cy);

                /* Check for significant mouse movement */

                do{
                        m_posbut(&nx,&ny,&bl,&br);
            }
                while(nx==mx && bl==0);

                /* If significant blank display for update */

                grrtulc(cx-8,cy-8,32,8,bg);

                /* Exit loop if left button pressed */
        }
        while(bl==0);

        /* Write final value selected to EGA */

        cval=itoa(val,bufv,10);

        if(val<0)
                fhatsay(0,cval,m[7],cx-8,cy);
        else
                fhatsay(0,cval,m[7],cx,cy);

        /* Return value selected */

        return(val);
 }
```

A.3 Code Fragment 3 - The Menu

```
/* Global variables defining menu options */

int num_opt;
char menu_title[20];
char menu_sub[20];
char menu_name[10][20];
char menu_desc[10][80];

menu(m)
int m[];

/* General purpose menu routine for selection of options */
/* using mouse driven cursor and graphics style menu */

{
        int i,mx,my,bl,br,nx,ny,tx,ty;

        /* Define Options */

        menu_options();

        /* Initialise EGA card and fonts */

        setega();
        initgraf(16,0,0);
        fontld(0,"IBMROM");

        grrtulc(0,0,639,349,m[0]);
        grrtul(0,0,639,349,m[4]);
        grrtul(0,330,639,330,m[4]);

        do{
                /* Draw menu boxes */

                grrtulc(216,104,200,num_opt*10+70,m[1]);
                grrtulc(220,100,200,num_opt*10+70,m[2]);
                grrtul(220,100,200,num_opt*10+70,m[4]);

                grrtulc(228,107,180,40,m[3]);
                grrtulc(230,105,180,40,m[2]);
                grrtul(230,105,180,40,m[4]);

                /* Titles */

                fhatsay(0,menu_title,m[5],320-(strlen(menu_title))*4,120);
                fhatsay(0,menu_sub,m[6],320-(strlen(menu_sub))*4,135);

                /* Options */

                for(i=0;i<num_opt+1,i++){

                        fhatsay(0,menu_name,m[7],230,160);
                }

                /* Install mouse and set movement mask */

                m_instal();
                m_horiz(220,420);
                m_vert(152,num_opt*10+60);
                m_curon();

                do{
                        /* Determine mouse position and button status */

                        m_posbut(&mx,&my,&bl,&br);

                        /* Evaluate to option */

                        my=(my-152)/10;

                        /* Expand option in message line */

                        fhatsay(0,menu_desc,m[7],10,345);

                        /* Check for movement */

                        do{
                                m_posbut(&tx,&ty,&bl,&br);
```

```
                    ty=(ty-152)/10;
                }
                while(ty==my && bl==0);

                /* If significant blank EGA display message line */

                grrtulc(5,332,630,15,m[0]);

                /* If left button pressed exit loop */
            }
            while(bl==0);

            /* Turn mouse cursor off */

            m_curoff();

            /* Perform selected function */

            option(my);

            /* If option selected is return exit loop */
        }
        while(my<num_opt);

        return;
}

option(opt)
int opt;
{
        /* Perform selected option */

        if(opt==0)
            option_1();

        else if(opt==1)
            option_2();

        else if(opt==2)
            option_3();

        else if(opt==3)
            option_4();

        else
            return;
}

menu_options()
{
        num_opt=4;

        /* Name and describe options */

        sprintf(menu_name[0],"Option A\0");
        sprintf(menu_desc[0],"Description of Option A's function\0");

        sprintf(menu_name[1],"Option B\0");
        sprintf(menu_desc[1],"Description of Option B's function\0");

        sprintf(menu_name[2],"Option C\0");
        sprintf(menu_desc[2],"Description of Option C's function\0");

        sprintf(menu_name[3],"Option D\0");
        sprintf(menu_desc[3],"Description of Option D's function\0");

        /* Last option to return */

        sprintf(menu_name[4],"Return\0");
        sprintf(menu_desc[4],"Return to Last Menu\0");
}
```

A.4 Code Fragment 4 - The Keyboard Handler

```
sinput(nch,cx,cy,str,lns,m,bg)
int nch,cx,cy,lns,m[],bg;
char str[];

/* Routine to input character string into a character array */
/* plotting on the EGA screen and returning string */

{
      int i,x;
      char inc[2];

      /* Write string to EGA (edit existing string length-lns) */

      i=lns;
      x=cx+lns*8;

      fhatsay(0,str,m[6],cx,cy);

      /* Start loop to read string from keyboard */

      do{
            /* Read character from keyboard and terminate */

            inc[0]=getch();
            inc[1]='\0';

            /* Check for valid characters */

            if(inc[0]==8){

                  /* If character is backspace delete character */

                  if(x>cx){

                        grrtulc(x-8,cy-7,8,8,bg);

                        /* Decrement element and x position counter */

                        i=i-1;
                        x=x-8;
                  }
            }
            else if(inc[0]!=13){

                  /* If character is valid then plot on EGA */

                  fhatsay(0,inc,m[7],x,cy);

                  /* Copy to string */

                  str[i]=inc[0];

                  /* Increment element and x position counter */

                  i=i+1;
                  x=x+8;
            }

            /* If number of characters exceeded re-loop */

            if(i==nch){

                  do{
                        /* Read character from keyboard and terminate */

                        inc[0]=getch();
                        inc[1]='\0';

                        /* Check for valid characters */

                        if(inc[0]==8){

                              /* Backspace - delete character */

                              if(x>cx){

                                    grrtulc(x-8,cy-7,8,8,bg);

                                    /* Decrement element
                                    and x position counter */
```

```
                        i=i-1;
                        x=x-8;
                    }
                }

                /* If character not BS or Enter - beep */

                else if(inc[0]!=13)
                    putchar(7);

                /* Repeat until Enter pressed
                or characters deleted */
            }
            while(i>=nch && inc[0]!=13);
        }

        /* Repeat loop until Enter pressed */

    }
    while(inc[0]!=13);

    /* Terminate string and return */

    str[i]='\0';
}
```

A.5 Code Fragment 5 - Directory Search and File Selection

```
/* Global declarations */

int num_files;
char file_name[96][50]

select_file(sel_file,file_ext,m)
int m[];
char sel_file[],file_ext[];

/* Routine to search specified directory for files with */
/* a given extension and return selected filename */

{
	int i,ii,j,jj,n,nn,p,pr,errval;
	int x,y,xx,yy,fp,comp,seg,ln,lnf;
	int ix,iy,idx,idy,mx,my,mmx,mmy,fx,fy,ex;
	int bl,br,tx,ty,tn,col,res,ci,fex,pc4base;
	char *str1;
	static char cv[1],str2[30];
	char dr[2],dir[9];
	char drdir[20],ext[10];

	FILE *in;

	/* Check drive ready */

	if( drive_ok(dr,m) != 0 ) return(-1);

	/* Check Directory for existence */

	if( dir_ok(dir,m) != 0 ) return(-1);

	/* search file for given extension */

	num_files=0;

	sprintf(ext,"*.%s\0",file_ext);

	file_search(drdir,ext,FIL);

	if(num_files==0){

		grrtulc(5,332,630,15,m[0]);
		fhatsay(0,"No files found",m[5],10,345);

		putchar(7);

		return(-1);
	}

	/* Draw EGA screen for directory list */

	grline(180,20,180,330,m[4]);
	grrtulc(186,54,440,270,m[1]);
	grrtulc(190,50,440,270,m[2]);
	grrtul(190,50,440,270,m[4]);
	fhatsay(0,"Image File",m[5],350,60);

	/* Arrange list horizontally */

	for(j=0;j<=num_files/5;j++){

		for(i=0;i<=4;i++){

			fp=i+j*5;

			if(fp>96)
				break;

			if(fp<num_files){

				fx=xx=200+i*80;
				fy=yy=80+j*10;

				fhatsay(0,file_name[fp],m[6],xx,yy);
			}
		}
	}

	/* Select file */

	grrtulc(5,332,630,15,m[0]);
```

```
	fhatsay(0,
	"Move Cursor to select file, current :",m[7],10,345);

	m_instal();
	m_horiz(200,580);
	m_vert(72,2*num_files+80);
	m_setcur(240,76);
	m_curoff();

	ex=0;

	do{
		/* Determine mouse position */

		m_posbut(&mx,&my,&bl,&br);

		/* Evaluate to file number */

		mx=((mx-200)/80);
		my=((my-70)/10);

		tn=my*5+mx;

		/* Validate file number */

		if(tn>=num_files){
			tn=num_files-1;

			tx=fx;
			ty=fy;
		}
		else{
			tx=mx*80+200;
			ty=my*10+80;
		}

		/*	Highlight current file and write to message line */

		fhatsay(0,file_name[tn],m[5],tx,ty);
		fhatsay(0,file_name[tn],m[5],312,345);

		/* Check for significant movement and button presses */

		do{

			m_press(0,&bl,&mmx,&mmy);

			if(bl>0)
				ex=1;

			if(ex==0){
				m_posbut(&mmx,&mmy,&bl,&br);
				mmx=((mmx-200)/80);
				mmy=((mmy-70)/10);
				mmx=mmx*80+200;
				mmy=mmy*10+80;
			}
		}
		while(mmx==tx && mmy==ty && ex==0){

		/* If left button pressed file selected */

		if(ex==0){
			grrtulc(360,332,200,15,m[0]);
			grrtulc(tx,ty-8,70,8,m[2]);
			fhatsay(0,file_name[tn],m[6],tx,ty);
		}
	}
	while(ex==0);

	/* return full file specification */

	sprintf(sel_file,"%s:\\%s\\%s.%s\0",dr,dir,file_name[tn],file_ext);

	strupr(sel_file);

	return(0);
}

file_search(dir,str,type)

/* Routine to transfer file found in Disk Transfer Area */
/* to global name array and increment files found */
```

```
int type;
char *dir,*str;
{
      int i,flag;
      char full_name[80],dta[43];

      sprintf(full_name,"%s\\%s",dir,str);
      set_dta(dta);

      flag=FIRST;

      while(search(full_name,flag,type)){

            if(strcmp(".",dta+30)!=0 && strcmp("..",dta+30)!=0){

                  sprintf(file_name[nfiles],"%s\0",dta+30);
                  num_files++;
            }
            flag=NEXT;
      }
}

search(full_name,flag,type)

/* Directory search using DOS Find File and Find Next File */
/* Functions 0x4e & 0x4f, store name in Disk Transfer Area */

char *full_name;
int flag,type;
{
      union REGS regs;

      regs.h.ah = 0x4e + flag;
      regs.x.cx = type;

      regs.x.dx = (int)full_name;

      intdos(&regs,&regs);

      if(regs.x.cflag==1)
            return(FALSE);

      return(TRUE);
}

set_dta(ptr)
char *ptr;

/* Set the DTA to the local dta buffer */
{
      union REGS regs;

      regs.h.ah = 26;
      regs.x.dx = (int)ptr;

      intdos(&regs,&regs);
}
```

A.6 Code Fragment 6 - The Critical Error Handler

```
/* Include files */

#include<dos.h>
#include<stdio.h>

/* External declarations */

extern int near flag;

void handler(unsigned deverr,unsigned errcode,unsigned *devhdr)
{
      /* Return error code */

      flag=errcode & 0x00ff;
      _hardretn(1);
}
```

A.7 Code Fragment 7 - Check Drive Status

```
/* Include files */

#include<dos.h>
#include<stdio.h>
#include<fcntl.h>
#include<sys\types.h>
#include<sys\stat.h>
#include<io.h>

/* Global and external declarations */

int near flag=0;
extern void handler(unsigned a,unsigned b,int c);

/* Main routine */

main()
{
        int m[8];

        /* Load critical error handler using Microsoft C function */
        /* DOS Int 0x24 */

        _harderr(handler);

        /* Check drive */

        drive_ok("A\0",m);
}

drive_ok(dr,m)

/* Check drive ready using critical error handler, Int 24h */

int m[];
char dr[];
{
        char a,chdr[10];

        FILE *in;

        /* Create imaginary filename */

        sprintf(chdr,"%s:$\0",dr);

        do{
            flag=0;

            /* Attempt to open file */

            in=fopen(chdr,"r");

            if(flag!=0){

                grrtulc(5,332,630,15,m[0]);
                fhatsay(0,"Drive Not Ready - Abort, Retry "
                ,m[5],10,345);

                putchar(7);

                a=getch();

                fhatsay(0,a,m[5],270,345);

                if(a=='a' || a=='A') return(-1);
            }
        }
        while(flag!=0);

        /* Repeat until operation aborted or succesful */

        /* If opened, close file and remove */

        fclose(in);

        remove(chdr);

        return(0);
}
```

A.8 Code Fragment 8 - Printer Port Status

```
/* Global definitions */

#define PRN_INT 0x17
#define STAT_RQ 0x02

printer_ok(m)

/* Routine to request and decode printer port status */

int m[];
{
        int i,stat;
        char a;

        union REGS regs;

        do{
             /* Set calling registers */

             regs.h.ah = STAT_RQ;
             regs.x.dx = 0;

             /* Execute Interrupt 17h */

             int86(PRN_INT,&regs,&regs);
             stat = regs.h.ah;

             /* Analyse bits set */

             if( (stat & 0x10) != 0 ){

                  putchar(7);

                  grrtulc(5,332,630,15,m[0]);
                  fhatsay(0,"Printer not Ready : ",m[5],10,345);

                  if( (stat & 0x01) == 1 )
                       fhatsay(0,"Check Power On - Abort, Retry"
                       ,m[5],170,345);

                  else if( (stat & 0x04) == 1 )
                       fhatsay(0,"Paper Out - Abort, Retry"
                       ,m[5],170,345);

                  else if( (stat & 0x04) == 0 && (stat & 0x01) == 0 )
                       fhatsay(0,"Printer Off Line - Abort, Retry"
                       ,m[5],170,345);

                  a=getch();

                  if(a=='a' || a=='A') return(-1);
             }
             else
                  return(0);
        }
        while(a=='r' || a=='R');

        /* repeat while 'retry' selected */

        return(0);
}
```

A.9 Code Fragment 9 - Run Length Encoding and Decoding

```
#define MAXI 36864L
#define MAXB 65530L

unsigned char huge pix[MAXI];
unsigned char huge line[MAXB];

run_length_encode(x,y,dx,dy,comp,fname)
int x,y,dx,dy,comp;
char fname[];
{
        int i,j,val,nv,si,nw;
        unsigned long int k,ios,tbr,tbw;
        unsigned int jj,np,nr,ly,res;
        char a;

        int out;

        /* Initialise board and component variables */

        pcinit();
        sscomp(comp);
        sorg(x+128,y+32);
        sdspmd(comp);

        /* Open file for low level, un-buffered write */

        out=open(fname,O_WRONLY|O_TRUNC|O_CREAT|O_BINARY,S_IREAD|S_IWRITE);

        /* Initialise sum variables */

        nr=dy/16;
        k=0;
        tbw=0L;
        tbr=0L;

        for(jj=0;jj<16;jj++){

              /* Read block of image data */

              movea(0,jj*nr);
              frim(dx,nr,pix);

              /* Encode block of data line by line */

              ios=0;

              for(j=0;j<nr;j++){

                    si=0;

                    do{
                          nv=0;
                          val=pix[si+ios];

                          do{
                                nv++;
                                si++;
                          }
                          while(pix[si+ios]==val && nv<256 && si<dx);

                          line[k++]=val;
                          line[k++]=nv-1;
                    }
                    while(si<dx);

                    ios=ios+dx;

                    /* Check size of buffer used */

                    if(k >= MAXB-2*dx){

                          /* Check for data compression */

                          tbw=tbw+k;

                          if(tbw>tbr+j*dx){

                                /* If not compressing - abort process */

                                printf("Data not Compressing - Aborting");

                                close(out);
```

```
                        remove(str);

                        return(-1);
                    }

                    /* Write compressed data, and zero buffer offset */

                    res=write(out,line,k);

                    if(res!=k){

                        printf("Disk Full / Disk Write Error");
                        close(out);
                        remove(str);

                        return(-1);
                    }
                    k=0;
                }
            }
            tbr=tbr+np;
        }

        /* Check for odd size image */

        ly=nr*16;

        if(dy>ly){

            /* and encode last image block */

            nr=dy-ly;
            np=nr*dx;

            movea(0,ly);
            frim(dx,nr,pix);

            ios=0;

            for(j=0;j<nr;j++){

                si=0;

                do{
                    nv=0;
                    val=pix[si+ios];

                    do{
                        nv++;
                        si++;
                    }
                    while(pix[si]==val && nv<256 && si<dx);

                    line[k++]=val;
                    line[k++]=nv-1;
                }
                while(si<dx);

                ios=ios+dx;

                if(k >= MAXB-2*dx){

                    tbw=tbw+k;

                    if(tbw>tbr+j*dx){

                        printf("Data not Compressing - Aborting");

                        close(out);
                        remove(str);

                        return(-1);
                    }

                    res=write(out,line,k);

                    if(res!=k){
                        printf("Disk Full / Disk Write Error");

                        close(out);
                        remove(str);

                        return(-1);
                    }
```

```
                    k=0;
                }
            }
            tbr=tbr+np;
        }

        /* Write encoded buffer contents */

        if(k>0){

            res=write(out,line,k);

            if(res!=k){
                printf("Disk Full / Disk Write Error");

                close(out);
                remove(str);

                return(-1);
            }
        }

        close(out);
        sorg(0,0);

        return(0);
}
```

```
run_length_decode(x,y,dx,dy,comp,fname)
int x,y,dx,dy,comp;
char fname[];
{
        unsigned int i,j,k,l,val,nv,nr,ly;
        long int np,nb,br,mb,nbr,nbl,btr;

        int in;

        /* Initialise board and component variables */

        pcinit();
        sorg(x+128,y+32);
        sdspmd(comp);

        /* Check size of file */

        getfsize(fname,&nb);

        /* Open file for low level, un-buffered read */

        in=open(fname,O_RDONLY|O_BINARY);

        /* Set image block size */

        nr=dy/16;
        np=nr*dx;

        if(nb<MAXB){

              /* If file smaller than allocated buffer read entire file */

              br=read(in,line,nb);

              l=0;
              j=0;

              do{
                    i=0;

                    do{
                          /* Read byte pairs */

                          val=line[l++];
                          nv=line[l++] +1;

                          /* Unpack into image buffer */

                          for(k=i;k<i+nv;k++){

                                pix[k]=val;
                          }

                          i=i+nv;

                          /* If image buffer full */

                          if(i==np){

                                /* Plot image and zero offset */

                                sdcomp(comp);
                                movea(0,j*nr);
                                fwim(dx,nr,pix);

                                i=0;
                                j++;
                          }
                    }
                    while(i<np);
              }
              while(j<16);
        }
        else{

              /* If larger than allocated buffer read in segments */

              br=read(in,line,MAXB);

              /* Determine bytes left to read */

              nbr=br;
              nbl=nb-nbr;
```

```
			if(nbl<MAXB)
				btr=nbl;
			else
				btr=MAXB;

			l=0;
			j=0;

			do{
				i=0;

				do{
					if(l==br && nbr<nb){

						/* If all bytes unpacked read next segment */

						br=read(in,line,btr);

						/* Determine bytes left to read */

						nbr=nbr+nb;
						nbl=nb-nbr;

						if(nbl<MAXB)
							btr=nbl;
						else
							btr=MAXB;
						l=0;
					}

					/* Read byte pairs */

					val=line[l++];
					nv=line[l++] +1;

					/* Unpack into image buffer */

					for(k=i;k<i+nv;k++){

						pix[k]=val;
					}
					i=i+nv;

					/* If image buffer full */

					if(i==np){

						/* Plot image and zero buffer */

						sdcomp(comp);
						movea(0,j*nr);
						fwim(dx,nr,pix);

						i=0;
						j++;
					}
				}
				while(i<np);
			}
			while(j<16);
		}

		close(in);

		sorg(0,0);
		return(0);
}

getfsize(str,fsize)
long *fsize;
char str[];
{
		struct find_t ff;

		/* Get file size using DOS Find File and transfer of size to DTA */

		_dos_findfirst(str,_A_NORMAL,&ff);

		*fsize = ff.size;
}
```

A.10 Code Fragment 10 - Quadtree Encoding and Decoding

```
#define MAXB 8192L

unsigned long int huge wval[MAXB];

quadtree_encode(x,y,dx,dy,comp,fname)
int x,y,dx,dy,comp;
char fname[];
{
        unsigned long int fos[12],wcnt,nbr,br;
        unsigned long int li,lj,lil,ljl;
        unsigned long int xmask,ymask,vmask;
        int mxy,i,j,pitch,col,siz;
        unsigned char img0,img1,img2,img3;

        FILE *out;

        /* Initialise board and component variables */

        pcinit();
        sscomp(comp);
        sdcomp(comp);
        sorg(x+128,y+32);
        sdspmd(comp);

        col=255;
        sfcolc(comp,col);

        /* Open file for buffered write */

        out=fopen(fname,"wb");

        /* Write initial resolution offset counters */

        for(i=0;i<12;i++){

             fos[i]=0L;
        }
        fwrite(fos,4,12,out);

        /* Define bit masks for elements of 32 bit word */
        /*  0- 7 = Pixel Value */
        /*  8-19 = Horizontal position of bottom left corner */
        /* 20-31 = Vertical position of bottom left corner */

        ymask=0xfff00000L;
        xmask=0x000fff00L;
        vmask=0x000000ffL;

        /* Initialise compression variables */

        nbr=12L;
        br=0L;

        wcnt=0;
        pitch=2;

        mxy=dx;

        if ( dy > mxy ) mxy=dy;

        /* Alter size of block until equal to image size */

        while (pitch<=mxy) {

             for (j=0;j<dy;j=j+pitch) {

                  for(i=0;i<dx;i=i+pitch) {

                       /* Read pixels at corners of block */

                       img0=rpix(i,j);
                       img2=rpix(i,j+pitch-1);
                       img1=rpix(i+pitch-1,j);
                       img3=rpix(i+pitch-1,j+pitch-1);

                       /* If all 4 pixels not the same */

                       if((img0!=img1)||(img0!=img2)||(img0!=img3)) {

                            /* Write block details to buffer - */

                            li=(long)i;
```

```
                    lj=(long)j;
                    li1=(long)(i+pitch/2);
                    lj1=(long)(j+pitch/2);

                    /* - if not already defined */

                    if(img0 != col) {

                          wval[br]=(lj<<20)+(li<<8)+img0;
                          fos[wcnt]++;
                          nbr++;
                          br++;
                    }
                    if(img1 != col) {

                          wval[br]=(lj<<20)+(li1<<8)+img1;
                          fos[wcnt]++;
                          nbr++;
                          br++;
                    }
                    if(img2 != col) {

                          wval[br]=(lj1<<20)+(li<<8)+img2;
                          fos[wcnt]++;
                          nbr++;
                          br++;
                    }
                    if(img3 != col) {

                          wval[br]=(lj1<<20)+(li1<<8)+img3;
                          fos[wcnt]++;
                          nbr++;
                          br++;
                    }

                    /* Flag block as already defined on image */

                    movea(i,j);
                    rfill(pitch,pitch);
               }
          }

          /* Check for buffer overflow */

          if(br>MAXB-4096){

               /* Write buffer and zero offset */

               fwrite(wval,4,br,out);
               br=0;
          }
     }

     /* Increase block size by factor of 2 */

     pitch=pitch*2;
     wcnt++;
}

/* Write contents of buffer */

if(br>0)
     fwrite(wval,4,br,out);

/* Write offset of each block resolution set */

fseek(out,0L,SEEK_SET);

fwrite(fos,4,12,out);

fclose(out);
nbr=nbr*4;

sorg(0,0);
}
```

```
quadtree_decode(x,y,dx,dy,comp,fname)
int x,y,dx,dy,comp;
char fname[];
{
	unsigned long int fos[12],wcnt,nbr,br,tbr;
	unsigned long int os[12],tos;
	unsigned long int xmask,ymask,vmask;
	int i,j,col,px,py,siz;

	FILE *in;

	/* Initialise board and component variables */

	pcinit();
	sorg(128+x,32+y);
	sscomp(comp);
	sdcomp(comp);
	sdspmd(comp);

	/* Define bit masks for elements of 32 bit word */
	/*  0- 7 = Pixel Value */
	/*  8-19 = Horizontal position of bottom left corner */
	/* 20-31 = Vertical position of bottom left corner */

	ymask=0xfff00000L;
	xmask=0x000fff00L;
	vmask=0x000000ffL;

	/* Open file for buffered read */

	in=fopen(fname,"rb");

	/* Read block resolution offset counters */

	fread(fos,4,12,in);

	/* Translate to file offsets */

	tos=48L;
	os[0]=tos;

	for(i=1;i<12;i++){
		tos=tos+fos[i-1]*4;
		os[i]=tos;
	}

	/*  Initialise block size to a pixel */

	siz=1;

	for(i=0;i<12;i++){

		/* Seek to start of block size list */

		fseek(in,os[i],SEEK_SET);

		/* If number of block size is less than allocated buffer */

		if(fos[i]<MAXB){

			/* Read in block words */

			fread(wval,4,fos[i],in);

			for(j=0;j<fos[i];j++){

				/* Unpack to determine value and position */

				py=(int)((wval[j] & ymask)>>20);
				px=(int)((wval[j] & xmask)>>8);
				col=(int)(wval[j] & vmask);

				/* and plot block */

				sfcolc(comp,col);
				movea(px,py);
				rfill(siz,siz);
			}
		}
		else{

			/* Read in first segment of block words */

			br=0;
			nbr=0;
			tbr=fos[i];
```

```
                do{

                     if(tbr>MAXB)
                          br=fread(wval,4,MAXB,in);
                     else
                          br=fread(wval,4,tbr,in);

                     /* Determine words left to read */

                     nbr=nbr+br;
                     tbr=fos[i]-nbr;

                     for(j=0;j<br;j++){

                          /* Unpack to determine value and position */

                          py=(int)((wval[j] & ymask)>>20);
                          px=(int)((wval[j] & xmask)>>8);
                          col=(int)(wval[j] & vmask);

                          /* and plot block */

                          sfcolc(comp,col);
                          movea(px,py);
                          rfill(siz,siz);
                     }
                }
                while(tbr>0);
           }

           /* Increase block size by factor of 2 */

           siz=siz*2;
      }

      fclose(in);

      sorg(0,0);
}
```

APPENDIX B

Publications

PUBLICATIONS

1. **"River Basin Surveillance by Remote Sensing"**, with T.R.E.Chidley, W.G.Collins, *EARSeL Syposium : Looking Back/Looking Forward - European Remote Sensing Needs in the 1990s*, 4-8th May 1987, Noordwijk, Holland.

2. **"An IBM PC-AT based Image Processing System"**, with T.R.E.Chidley, *The Remote Sensing Society Workshop on Use of Small Computers for Image Processing in Education*, 20th May 1987, Nottingham University, Nottingham, UK.

3. **"Low Cost Techniques for Machine Processing and Integration of Image and Map Data"**, with M.R.Pooley, T.R.E.Chidley, W.G.Collins, *Remote Sensing '87*, 17-20th August 1987, Prince of Songkhla University, Hat Yai, Thailand.

4. **"Applications of Integrated Image and Map Data with Reference to Land Evaluation in Thailand"**, with M.R.Pooley, T.R.E.Chidley, W.G.Collins, *Remote Sensing '87*, 17-20th August 1987, Prince of Songkhla University, Hat Yai, Thailand.

5. **"River Basin Surveillance using Remotely Sensed Data"**, with T.R.E.Chidley, *Advances in Digital Image Processing - 13th Annual Conference of the Remote Sensing Society*, 7-11th September 1987, Nottingham University, Nottingham, UK.

6. **"An Evaluation of Low Cost Remote Sensing Techniques for Water Resources Studies"**, Report produced for the National Remote Sensing Centre, November 1987.

7. **"River Basin Surveillance"**, *Annual British Hydrological Society Syposium*, 6th December 1987, Aston University, Birmingham, UK.

8. **"Low Cost Remote Sensing Techniques for Developing Countries"**, with T.R.E.Chidley, J.Elgy, *EARSeL News*, No.35, March 1988.

9. **"Cutting the Price of Remote Sensing"**, with T.R.E.Chidley, J.Elgy, *Spectrum - British Science News*, No.216/1988/SPECTRUM/8, 1988.

10. **"The Geological Workstation of Today : with some thoughts for Tomorrow"**, with T.R.E.Chidley, R.B.Oldfield, *The Geological Workstation of Tomorrow*, 11th May 1988, British Geological Survey, Keyworth, UK.

11. **"Digital Elevation Models and their Application in Remote Sensing of Water Resources"**, with T.R.E.Chidley, *IGARSS'88*, 13-16th September 1988, Edinburgh University, Edinburgh, UK.

12. **"Desktop GIS in Land and Water Resource Development"**, with T.R.E.Chidley, *Remote Sensing Society Workshop on Desktop GIS*, 24th May 1989, Nottingham University, Nottingham, UK.

13. **"Some Advances in Videographic Remote Sensing"**, with T.R.E.Chidley, *IGARSS'89*, 10-14th July 1989, Vancouver, Canada.

14. **"New Possibilities for Precipitation Estimation for River Basin Managers in Developing Countries"**, with T.R.E.Chidley, A.N.Siyyid, *Radar Applications in Meteorology*, 20th July 1989, Salford University, Manchester, UK.

15. **"A River Basin Information System for Developing Countries"**, with T.R.E.Chidley, D.G.Evans, *Remote Sensing for Operational Applications - 15th Annual Conference of the Remote Sensing Society*, 13-15th September 1989, Bristol University, Bristol, UK.

16. **"Videographic Mapping for Strip Feature Monitoring"**, with T.R.E.Chidley, W.G.Collins, *Remote Sensing for Operational Applications - 15th Annual Conference of the Remote Sensing Society*, 13-15th September 1989, Bristol University, Bristol, UK.