



Durham E-Theses

Raster data structures and topographic data

Adams, Timothy A.

How to cite:

Adams, Timothy A. (1982) *Raster data structures and topographic data*, Durham theses, Durham University. Available at Durham E-Theses Online: <http://etheses.dur.ac.uk/7689/>

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

University of Durham
Department of Geography

Raster data structures and topographic data

A thesis submitted for the degree
of Doctor of Philosophy
in the Faculty of Science

by

Timothy A. Adams B.Sc., M.Sc.
Graduate Society

June 1982

The copyright of this thesis rests with the author.
No quotation from it should be published without
his prior written consent and information derived
from it should be acknowledged.



ABSTRACT

The use of computers to assist in map-making has been growing for two decades; their speed of operation, large data storage capacity and flexibility of usage have been major factors in establishing many development and working computer mapping systems throughout the world. In Britain, the Ordnance Survey has supported a digital solution to the production, storage and display of large scale topographic maps since 1972. Until now, the work of the Ordnance Survey - and, indeed, most topographic map-makers in Britain who are investigating digital techniques - have adopted a vector-based strategy to digital mapping in which the data are held as a series of coordinate-points describing the lines shown on the map images. Comparatively little work has been undertaken in Britain on the use of raster-based methods of data capture and storage in which map images are resolved into arrays of small cells or picture elements by appropriately tuned scanning devices. This alternative strategy is known - from work carried out in other countries, chiefly the United States - to be suitable for some types of data manipulation, although its suitability for Ordnance Survey mapping applications is unknown. Very little investigation has been made anywhere in the world of the manipulation of raster data structures by the recently developed array processor computers; almost all existing work is restricted to the use of traditional serial machines.

This thesis reports on a three year study carried out in the University of Durham to investigate the applicability of raster data processing for the work of the British national mapping organisation. In particular, it describes the distinction between vector and raster applications with geographic data and the likely characteristics of suitable raster data structures on both serial and parallel computers. A section is also included which describes the nature of scanning trials carried out on a number of commercial devices; it has thus been possible to assess not only the likely advantages and limitations of handling British large scale map data in raster form but also its technical feasibility. The work reports on the likely volumes of data to be expected and describes parallel algorithms for operations such as polygon creation (and, indirectly, the creation of node and link vector files).

ACKNOWLEDGEMENTS

I am indebted to many individuals and organisations who gave freely of their time and resources to enable me to undertake this research study. Chief amongst them are:

The Ordnance Survey and Science and Engineering Research Council for joint financial support under the CASE scheme.

Professor D.W. Rhind who has inspired and supervised me throughout the entire project, providing much patience, concern and direction over the last three years.

Brigadier C.N. Thompson, Col. S.E.G. Fraser, Messers. P.W. Wesley, I.T. Logan, P.E. Haywood and N.S. Smith of the Ordnance Survey for valuable help and advice on survey, mapping and OS-related technical matters.

Dr. J.F. Harris (Oxford University), Mr. C. Lloyd (Kongsberg Ltd.), Dr. W. Pferd (Bell Laboratories), Mr. D.G. Smith (USGS) and Mr. R. Brown (Versatec Electronics) who provided real assistance on the use of their respective systems with data specific to the needs of this research.

Dr. R. Harris of the Geography Department, University of Durham, for lengthy discussions on all aspects of the work and for reading preliminary drafts of the thesis text.

Messers. M. Munro and N.F. Hall of the Computer Unit at the University of Durham for advice on all aspects of serial computer programming, text-formatting and graphics.

Mr. K.A. Smith of the DAP Support Unit at Queen Mary College, London for many hours of help with parallel algorithms and DAP programming.

Mr. D. Hudspeth who produced the photo-reductions of my computer plots for inclusion as figures in the thesis.

My final and perhaps greatest debt is to my fiancée, family and friends from Scunthorpe for their seemingly endless encouragement during the last seven years.

Timothy A. Adams
June 1982

LIST OF CONTENTS

1	THE DIGITAL HANDLING OF TOPOGRAPHIC INFORMATION	1
1.1	The need for topographic mapping	1
1.2	A digital approach to topographic mapping	4
1.2.1	The benefits of digital mapping	6
1.3	User requirements of topographic maps	9
1.4	The role of the Ordnance Survey	11
1.5	The objectives of this thesis	16
2	VECTOR DATA STRUCTURES	17
2.1	Introduction	17
2.1.1	The view of the United Kingdom	18
2.2	Spatial data	19
2.2.1	The incorporation of attributes	20
2.2.2	Data capture	21
2.3	Digitising procedures	26
2.3.1	Data validation	28
2.4	Data organisation and storage	31
2.4.1	A suitable data structure	33
2.4.2	Data characteristics	38
2.5	The digital mapping approach of Ordnance Survey	44
2.5.1	Preparation stage	48
2.5.2	Digitising stage	49

2.5.3	Edit stage	51
2.5.4	Examination and archiving stage	53
2.5.5	Cartographic enhancement stage	54
2.6	The need for topology in the Ordnance Survey data structure	56
2.7	The restructuring project for Ordnance Survey digital data	58
2.7.1	The file structure of the PMA restructuring process	60
2.7.2	Uses of the restructured low-level data	63
2.8	The use of fast-digitising systems at Ordnance Survey	66
2.9	Data base management systems and topographic data	70
2.10	The United States Geological Survey alternative for vector-based digital mapping	75
2.10.1	The role of the National Mapping Division of USGS and the National Mapping Program	77
2.10.2	The data-structure to be adopted	79
2.11	Conclusions	82
	Appendix 2.1	83
3	RASTER DATA STRUCTURES	87
3.1	Introduction	87
3.2	Raster data representations	89
3.3	The data capture stage	96
3.3.1	The drum-based principle of operation	97

3.3.2	The flat-bed principle of operation	98
3.3.3	Appropriate pixel resolution	100
3.4	The implementation of a raster-based data structure	102
3.4.1	Data compaction methods	103
3.5	Some examples of raster-based applications other than ^{topographic} mapping	111
3.5.1	Grid square-based census mapping	111
3.5.2	Handling multispectral scanner data	116
3.6	The revision of a raster-based topographic data structure	120
3.7	Raster data handling appropriate to map data	122
3.7.1	Basic image analysis for feature recognition	127
3.8	User experience	133
3.8.1	A United States view	135
3.8.2	The European view	138
3.9	Conclusions	141
4	COMPARISONS OF RASTER AND VECTOR REPRESENTATIONS	143
4.1	Introduction	143
4.2	The basic characteristics of data handling	146
4.2.1	Data entry	146
4.2.2	Data manipulation	160
4.2.3	Data presentation	169
4.2.4	Data storage	174
4.3	Summary and conclusions	176

5	THE MANIPULATION OF RASTER DATA ON A SERIAL COMPUTER . . .	179
5.1	Introduction	179
5.2	A serial-type computing environment	184
5.3	The development of a vector-to-raster conversion system	187
5.3.1	Design aims	188
5.3.2	Theory	192
5.3.3	Some data transfer considerations	193
5.3.4	Development of the vector-to-raster conversion algorithm	196
5.3.5	Development of associated utility programs	199
5.3.6	Software tuning	203
5.4	Data compaction techniques for the raster image	210
5.4.1	Development of a coded-delta compaction algorithm	213
5.4.2	Software tuning	220
5.5	The choice of pixel resolution	222
5.6	Work applications related to OS requirements	232
5.6.1	Some applications toward link and node detection	233
5.6.2	Attempts to rebuild the raster data image into a vector file	235
5.7	Conclusions	242
	Appendix 5.1	245

6	THE MANIPULATION OF RASTER DATA ON A PARALLEL COMPUTER	248
6.1	Introduction	248
6.2	An array-type computing environment	250
6.2.1	The vector or pipeline array processor	251
6.2.2	The Distributed or cellular Array Processor	257
6.3	A raster-based data structure appropriate to the ICL DAP	270
6.3.1	The transfer of data to the ICL 2980	274
6.3.2	Data reformatting on the DAP	278
6.4	Raster-based manipulation appropriate to an array-processor	279
6.4.1	DAP-assisted raster graphics - an example of a parallel image processing algorithm	281
6.4.2	The automated addition of topology to a raster image	290
6.5	Conclusions	310
	Appendix 6.1	313
7	AN APPRAISAL OF RASTER-DRIVEN HARDWARE	316
7.1	Introduction	316
7.2	Scanning tests of Ordnance Survey large scale documents	319
7.2.1	The role of a raster-driven data entry system	323
7.3	Descriptions of commercial raster-scanning systems	327
7.3.1	Broomall Industries Inc.	327

7.3.2	Bell Telephone Laboratories Inc.	330
7.3.3	Kongsberg Ltd.	332
7.3.4	Oxford University Image Analysis Group . . .	336
7.3.5	Scitex Corporation	338
7.3.6	Other scanning devices	340
7.4	An appraisal of existing raster systems	343
7.4.1	The characteristics of the raster files . .	346
7.4.2	The characteristics of the raster-to-vector conversions	364
7.5	Descriptions of raster-driven output devices . . .	400
7.5.1	Raster-driven refresh graphics displays . .	401
7.5.2	Raster-driven hard-copy plotters	404
7.6	Conclusions	409
	Appendix 7.1	416
8	CONCLUSIONS AND IMPLICATIONS FOR A NATIONAL DIGITAL RASTER TOPOGRAPHIC DATA BASE	419
8.1	Summary of the work	419
8.2	Implications from results of studies into the 'state of the art' of digitising systems	421
8.3	Implications from results of studies into post hoc manipulation of scanned data	426
8.4	Implications for the national topographic digital data base	431
8.4.1	A raster-based data base	435

8.4.2	The interrelation between vector and raster data in the data base	441
8.5	Suggested areas of further study	445
	BIBLIOGRAPHY	449
	LIST OF PERSONAL COMMUNICATIONS	463
	ANNEXURES TO CHAPTER 5 (software listings relating to text)	466
	ANNEXURES TO CHAPTER 6 (software listings relating to text)	486

N.B. Appendices to specific chapters appear at the end of each chapter.

Annexures comprise software listings and appear at the end of the thesis.

LIST OF FIGURES

<u>Figure</u>	<u>Description</u>	<u>Page</u>
1.1	Samples of OS DMC digital data produced with OS plotting software	13+
2.1	Types of moveable cursor on manual digitising tables	23
2.3	Description of the Ordnance Survey (DMB) large scales digital flowline	47
2.4	An example of OS data restructuring	59
2.5	The structure of a relational data base	70
2.6	The current OS restructured vector data structure expressed in relational terms	72
2.7	A tree structure to store link and node data	73
2.8	An example of the complex data base schema required to represent the simplest of vector link and node structures	74
2.9	The Ordnance Survey restructured vector data base	76
3.1	Some scanning strategies used by flat-bed scanners	100
3.2	Indexing with access vectors for jagged arrays	107
3.7	The distinction between vector and raster notation	123
3.8	The representation of points, lines and areas in raster mode	124
3.9	The effects of four and eight direction pixel connection	125

3.10	An example of a simple distance transformation to recognise a road carriageway on an OS 1:1250 map . . .	132
3.11	The use of distance transformations to detect a road junction in the OS 1:1250 data	133
4.1	Map feature delineation by nodal end-points	157
5.2	Centre-lines of roads on OS 1:1250 map shown as (a) crude raster by lineprinter (b) vectors on a pen-plotter	200 201
5.4	The use of one-byte variables for numerical data . . .	204
5.6	Example of a coded-delta compression scheme	213
5.11	The effect of pixel size on compaction times	226
5.12	The effect of pixel size on compaction efficiency . . .	227
5.13	Percentage reduction of raster image with pixel size	227
5.14	The effect of pixel size on percentage sparsity . . .	228
5.15	Percentage sparsity against percentage reduction of raster buffer	228
5.16	Section of a vector file rebuilt from a simulated raster image of OS sheet SO 9283SE	238
6.1	The basic components of a pipeline array processor . .	253
6.2	Simplified elements of the ICL Distributed Array Processor	259
6.3	A typical ICL 2900 configuration with Distributed Array Processor	262
6.4	Block diagram showing the modular nature and the separation between host- and DAP-related sections of typical DAP software	268

6.5	A hierarchical data structure for raster data on a cellular array processor	272
6.7	Magnetic tape format suitable for DAP entry	276
6.8	A block summary of the graphics routines developed for the DAP	284
6.9	The generation of an index appropriate to the level 12 data for an OS 1:1250 map sheet	286
6.10	A parallel algorithm to compress a 64 x 64 image to an 8 x 8	287
6.11	Raster display of two public buildings at three data scale levels	289
6.12	Lineprinter representation of two DAP planes whilst extracting sub-parcels from a master raster image of map data	295
6.13	Raster-to-vector conversion of sub-parcel boundaries before and after point filtering	302
6.14	Example of interactive display of Sigma raster screen showing sub-parcel references with associated spectral-type signatures	306
6.15	Types of pixel pattern to measure shape	307
6.16	The interconnection of pixel clusters to determine the shape of sub-parcel units	308
7.2	Extract of special OS source document used for scanning trials with text and stipple removed	321
7.3	The effect of incorrect grey-level thresholding with OS map data	325

7.6	Extracts of raster image of OS map generated by Bell Laboratories' EDAD scanner	347+
7.7	Extracts of raster image (grey-levels) of OS map generated by Kongsberg KartoScan scanner	351
7.8	Extracts of raster image generated by Scitex Corp. scanner at USGS	355+
7.9	Examples of vector data generated by Bell Laboratories conversion software from Bell raster data	367+
7.11	Examples of vector data generated by Kongsberg conversion software from Kongsberg raster data . . .	374+
7.12	Examples of vector data generated by Oxford University conversion software from Oxford PEPR raster data	380+
7.13	Examples of vector data generated by Oxford University conversion software from Scitex raster data	383+
7.14	Examples of vector data generated by USGS conversion software from Scitex raster data	389+
7.15	Bell Labs. vector data plotted on Ferranti Master plotter at OS	393
7.16	Kongsberg Ltd. vector data plotted on Ferranti Master plotter at OS	393
7.17	Oxford University/PEPR scanner vector data plotted on Ferranti Master plotter at OS	394
7.18	Oxford University/USGS Scitex scanner vector data plotted on Ferranti Master plotter at OS	394

7.19 USGS/Scitex scanner vector data (unfiltered)
plotted on Ferranti Master plotter at OS 395

7.20 USGS/Scitex scanner vector data (filtered)
plotted on Ferranti Master plotter at OS 395

7.22 Extract of a 4096 x 4096 simulated raster image
plotted on a Versatec electrostatic raster plotter . 406

7.23 Extract of a 8000 x 8000 Scitex raster image
plotted on a Versatec electrostatic raster plotter . 407

8.2 Suggested data structure for a UK raster
topographic data base 437

8.3 The basic components of a raster-based topographic
data base system using parallel processing
techniques 443

N.B. A '+' sign attached to page numbers shows figures
which overlap on to consecutive pages.

LIST OF TABLES

<u>Table</u>	<u>Description</u>	<u>Page</u>
2.2	Extract of codes for districts within counties within regions	37
3.3	Comparison of the bit-map and coded-delta schemas for zero suppression of 100 per cent population census data for Durham census county	109
3.4	Comparison of the bit-map and coded-delta schemas for zero suppression of OS sheet SO 9283SE held as a 4096 x 4096 raster byte map	110
3.5	Levels of index for the CRU data base of 1971 UK population census statistics by 1 kilometre grid-square	115
3.6	User options when requesting remote sensed data from the NASA EROS Data Centre at Sioux Falls	117+
5.1	Performance of the conversion stage for a 1:1250 scale vector file	195
5.3	Breakdown of a 5000 x 5000 raster image of NZ 2642NE by utility RASTER%	202
5.5	The effects of software tuning on the vector-to- raster conversion of OS 1:1250 data	209
5.7	Comparison of the central processor times required to read a complete raster image of 5000 x 5000 pixels, scan-line by scan-line	216
5.8	The effects of data compaction on two OS 1:1250 map sheets	218

5.9	A summary of the characteristics of each scan-line for sheet NZ 2642NE after coded-delta compaction	221
5.10	The effects of pixel resolution on the raster image of sheet NZ 2542NE	225
5.17	DEC PDP-10 process times to convert a 4096 x 4096 raster image of a 1:1250 sheet into vector form	239
6.6	A summary of the storage associated with the DAP hierarchical data structure	274
7.1	Summary of ^{Some} known existing raster-driven scanning systems	318+
7.4	The effect of scanning resolution on pixel swathes on the Kongsberg KartoScan	334
7.5	Summary of the characteristics of the raster files generated by four tested scanning systems	345
7.10	Summary of the characteristics of the vector files produced by the tested commercial systems	372
7.21	Summary of the processing times necessary for raster to vector conversion	398
8.1	Estimation of the likely size of the GB basic scale topographic data base	433

N.B. A '+' sign attached to page numbers shows tables
which overlap on to consecutive pages.

GLOSSARY OF ABBREVIATIONS USED IN THE TEXT

DAP	ICL Distributed Array Processor
DMA	Digital Mapping A - first digital flowline operated by the Ordnance Survey
DMB	Digital Mapping B - current digital flowline at Ordnance Survey. Replaced the DMA system in mid-1970's.
DMC	Data format for digital data supplied on magnetic tape to customers by Ordnance Survey
DoE	Department of the Environment
ECU	NERC Experimental Cartography Unit, Swindon
ETL	Engineering Topographic Laboratory, Fort Belvoir
GIMMS	Geographic Information Mapping and Manipulation System
GINIS	Graphic Interactive Information System. Data base management software by Kongsberg Ltd.
HMSO	Her Majesty's Stationery Office
IGES	Interactive Graphics Editing System developed by Laser-scan Laboratories Ltd.
MCE RE	Mapping and Charting Establishment of the Royal Engineers
MSD	Master Survey Document - master record of each basic scale map in GB
NERC	Natural Environmental Research Council
NG	National Grid
PEPR	Precision Encoding Pattern Recognition - High precision raster scanner used at Oxford University.
OS	Ordnance Survey
QMC	Queen Mary College, University of London
SERC	Science and Engineering Research Council
TODDS	Topographic Digital Development System - interactive digital system currently being tested by Ordnance Survey
USGS	United States Geological Survey

1 | THE DIGITAL HANDLING OF TOPOGRAPHIC INFORMATION

1.1.1 | THE NEED FOR TOPOGRAPHIC MAPPING

Over a time period spanning nineteen decades, the role of national topographic mapping in Britain has gained increasing importance, demanding a data content adequate and sufficiently flexible for a wide variety of uses. The design and creation of appropriate mapping series as a collection of map sheets on film or paper has thus had three major motivations: as a national archive for topographic information, as a method of display for this archive and as a base on which to link and display much other data such as soils, geology, census of population and other statistics.

Arguably, we have already or are collecting enough information concerning the surface of the earth and human activities on it. Almost all developed countries now provide topographic mapping through one or more government-owned and controlled organisations (Bomford 1979). In terms of human resources, well in excess of £2000 million is spent each year on survey and mapping (United Nations 1976). Yet, such heavy commitment to the collection and organisation of environmental data seems evidently justified; there is an

Chapter one



apparently unceasing practical requirement for more detailed, more up-to-date or more appropriately classified map series. Without them, planning strategies and changes in national and regional policy have to be made on hunches rather than with factual evidence. Thus, it is not so much the quality and amount of data being collected which is questionable, it is the strategies for their appropriate storage and manipulation which require concise definition and adequate investigation to serve the needs of a modern state.

Developments in the handling of appropriate map data in recent years have been rapid. On the more applied side of British cartography, two developments stand out as being of crucial importance in the 1970's. The first of these is the use of revised photo-mechanical procedures to revolutionise the creation and distribution of maps. In Britain, the Ordnance Survey (OS) is a largely centralised source of detailed topographic data; these are used for a wide variety of purposes (Drewitt 1973). A consequence of the detailed mapping coverage by the OS is a large number of map sheets at the basic scales (section 2.5). The cost of storage and distribution of such vast numbers of litho-printed sheets, allied with their rapid obsolescence in a system which is based on continuous revision of the map detail, ensures that they are increasingly unattractive in financial terms and, moreover, are bulky and difficult to handle. The OS solution

(Marles 1971) has been to transfer the basic scales (1:1250 and 1:2500) maps on to 35mm microfilm and to store each sheet as an insert on a punched card. Complete coverage of those areas of Great Britain served by 1:1250 and 1:2500 scale basic maps can now be held in this fashion in one large filing cabinet and copies printed immediately, on demand, by every main agent in the country.

Inevitably, such procedures as these have disadvantages (OS 1981b, Adams et al 1980): the image quality of the printed-on-demand large scale maps is not as high as lithographically printed sheets. In a recent study into OS revision procedures, an examination of the opinions of the customers of OS mapping products have shown an apparent loss of confidence in the standards achieved with the microfilming procedures (OS 1981b). The research team which performed this study recommended that the use of microfilming for data up-date and display should cease until improvements are made in the quality of the presentation of the data. Nonetheless, the creation and dissemination of the standard maps of Great Britain have, within a decade, been transformed by the use of these methods. Indeed, their very success - at least, for the short term - has been a contributory factor in slowing down the rapid adoption of the second of the major developments in British cartography - that of computer assistance for the production and storage of map data.

1.2 A DIGITAL APPROACH TO TOPOGRAPHIC MAPPING

The methodologies to store and manipulate map data by computer have been known for almost two decades. The principles on which 'automated cartography' are based are simple and were described in a workable form as early as 1964 by Bickmore and Boyle. Chapter two of this thesis describes the traditional digital method - at least outside some American military operations - for the digital portrayal of topographic features. To hold map data as a series of lines, each comprising numerous pairs of digital coordinate values - a vector strategy - has been commonplace; the preservation of linear entities within a computer data structure mimics the human understanding of a map image. Much more recently, attention has been focussed on alternatives to this now well-established approach. A cellular representation of spatial data, in which a map image is resolved into a series of contiguous (usually square) cells known as picture elements or pixels - a raster strategy - is seen as having much potential in this area of data organisation. Though the procedure has long been used in low-quality map production, notably via use of the computer line-printer (e.g. see Coppock 1975, Schmidt and Zafft 1975), developments in technology such as to permit high-quality reproduction on an economic and reliable basis have only occurred recently. Chapter three gives an introduction to the intricacies of

this approach to automation in the mapping industry and later chapters give accounts of investigations into its suitability for the handling of the basic scales mapping in Great Britain.

The advances of digital techniques in map production have apparently been inevitable; automation is now increasingly involved in all stages of the mapping process. Direct digital readout is available from field instruments (Palmer 1980), from stereo-plotters (Petrie 1981) and from satellites (Barrett and Curtis 1976). Almost all survey data reduction is done on calculators or mainframe and micro-computers (Walker and Whiting 1980) and plotting topographic data under computer control has been possible for at least a decade. Even the control of some lithographic printing processes is now carried-out by computer (e.g. Crosfield Electronics 1981). Reasons for this apparent rapid adoption of digital techniques are clear. The real costs of computing power have been falling by an order of magnitude every five years for at least fifteen years and this trend is expected to continue in the 1980's. Such a diminution in costs ensures access to computer resources by increasing numbers of people; it is anticipated that by 1985, some seventy per cent of the total US work force will have access to a computer (Lord Avebury 1978). Given all this, it is also clear that the majority of operations performed by mapping agencies (data capture,

processing, storage, presentation and transmission) are likely to be affected by the increasing rapid rate of technological change and in particular by the fall in computing costs.

1.2.1 The benefits of digital mapping

The many benefits claimed for the use of computers in map-making have been itemised and evaluated elsewhere (see Fraser Taylor 1980, Rhind 1977). The primary advantage of a digital approach to mapping is flexibility: the ability to produce maps whose scale, content and appearance is solely related to the needs of users has serious implications for the continued use of traditional litho-printed sheets. In essence, sheet edges should become transparent to the users of digital maps - the user can 'customise' the images being produced to suit his exact needs with features, area of interest and scale being selected by parameters supplied to the controlling software. A speed-up in production rates would also seem inevitable given the plotting speeds of existing computer hardware; plotters can operate at perhaps 1000 times the speed of the average draughtsman (Rhind 1977) and the use of sophisticated computer output on microfilm (COM) devices can improve upon these speeds even further (Adams et al 1980).

In principle, the production of maps by computer can be cheaper than with conventional processes although there is commercial evidence both to support (Leatherdale 1977) and to undermine (Thompson 1978) this hypothesis. Rhind (1977) pointed out the complexities of costing map-production; such an exercise is only possible over an extended period of time and is difficult to compare from institution to institution. An underlying factor which governs the cost-effectiveness of a digital strategy is the volume of data involved. Many national mapping agencies have discovered that the high data volumes with which they are concerned (Adams and Rhind 1981) can only be handled cost-effectively by the most sophisticated of methods which have been developed recently to enhance the speed and accuracy of the data capture and presentation processes (McEwen 1980, Thompson 1978). A description and appraisal of existing digital methods known to the author is given in chapters two and three.

The storage of a national topographic archive in digital form offers - in principle - greater security, timeliness and accuracy than holding it as a collection of printed sheets. Though much criticised (eg. by Robinson and Petchennik 1975), studies of information theory have shown that map images can act as extremely dense stores of information. Clearly, the life-expectancy of paper sheets, or indeed archived film positives, is not indefinite, and the effects of extremes of

temperature or humidity place serious risks on the security of the survey record. To hold the same data as alphanumeric characters in computer form provides for more security and flexibility; a series of numeric coordinates can always be recalled to their original accuracy and then rounded to suit the needs of the user. Each copy is then as good as the original - unlike the photographic copying procedure.

The maintenance and revision of the topographic archive is arduous for graphic data; every manual re-drawing is likely to continue and to compound errors. A digital strategy for map revision is only likely to be cost-effective if the data are already held in digital form; serious overheads of cost and time are inherent in the conversion processes. Yet, once held in a computer form, the map data can be archived and up-dated with relative ease to produce a more dependable system. Clearly, it is the conversion process from existing formats to a digital representation which is providing most concern at present and it is this area where most applied effort is being devoted in an attempt to resolve some of the operational difficulties.

1.3 USER REQUIREMENTS OF TOPOGRAPHIC MAPS

A number of recent market research studies to investigate the needs of users of topographic maps in Britain have described the difficulties in achieving any clear and concise result (OS 1981b, HMSO 1979). Essentially, only the highly motivated users tend to respond to the mechanisms of such studies, which in itself produces 'biased' results since an accurate study should reflect the opinions and requirements of all map users. Topographic maps are complex sources of information comprising a wide variety of individual features. Most users, however, do not associate with any particular feature which is most important to them but tend to seek a faithful and reliable representation of the ground at that time (OS 1981b). Generally, it is clear that most users expect up-to-date information presented in accordance with a set of agreed conventions. With particular reference to Britain, it is apparent that even the major users of OS mapping (e.g. Local Authorities and Public Utilities) are content to accept the services offered by the mapping agency when they might be better off with presenting a case to obtain a service tailored to their needs (HMSO 1979). Reasons for this apparent apathy are unclear, other than lack of time to prepare the case on the part of local Authority officers. Essentially, many map users are conservative and have not clearly defined their future needs (OS 1981b).

It does seem inevitable, however, that greater emphasis and importance will be placed in future upon geographical information systems (GIS) generally (Rhind 1976), although much operational experience is limited at present and hence a detailed definition of uses is difficult to achieve. The types of user requests to future digital topographic archives are bound to increase in complexity, as users focus their operational requirements and discover the potential benefits of computers in mapping. In order to provide adequate data quantity, with sufficient versatility, the GIS of the future will demand robust interfaces with topographic and much other data. This can only be achievable with clear interface specifications and (probably) significant skills from the computer programmers. Essentially, an interactive interface between the user and the data archive is a prime objective. With the use of a succinct dialogue, the user should be able to ask a set of questions which evoke a conversational, tabular or graphic response from the system. A typical request, for example, might be to produce a map of all hospitals in the Midlands, serving less than 50,000 people, which are located within a five mile radius of a motorway, and to demarcate on the map the shortest route between these hospitals and the motorway junctions. Clearly, such a request requires access to Area Health data and population data in addition to the base topographic information. Such complex interrogations as this, involving the

cross-referencing of many data variables in several formats, can - in theory - best be handled with sufficient speed and efficiency by a computer. Indeed, many of the problems of concatenating different data such as census enumeration districts with post-code sectors or local planning regions can only be performed sufficiently quickly by digital methods.

At present, it is unclear at what rate these more sophisticated GIS applications will develop; hence a basic problem exists - to what extent should the OS digital mapping be tailored to existing, in-house, map reproduction facilities or be orientated to multi-purpose digital topographic data bases. Adoption of a strategy based upon a short term viewpoint may well compromise future uses, yet the creation of data in sophisticated structures will certainly increase the short-run costs.

1.4 THE ROLE OF THE ORDNANCE SURVEY

Ordnance Survey is the national mapping agency in Britain. This country has arguably the best map coverage in the world, consisting of c. 220,000 basic scale (1:1250, 1:2500, 1:10000) maps and much smaller numbers of small (1:25000 (c. 2000), 1:50000 (204), 1:250000 (9)) scale maps, all

up-dated comparatively frequently (OS 1981a). The nature of mapping - especially the large scales - undertaken by the OS is much different to that undertaken by other world mapping agencies; the basic scale of mapping produced by the United States Geological Survey (USGS), for example, is 1:24000 scale which closely resembles the 1:25000 scale mapping of the OS. To achieve its operational programme, OS employs 3371 staff, of which 1171 are field surveyors and 1048 are cartographic draughtsmen (OS 1981a); comparative figures for map-making staff in the USGS are shown in section 2.10.1. At the present time, only a small percentage of the work force has any real training or expertise in computer methods (HMSO 1979) and hence a carefully controlled induction scheme is essential if any serious automation of the map-making process is to develop.

The future of the OS was reviewed recently by the Ordnance Survey Review Committee (or the Serpell Committee, named after its chairman). The report (HMSO 1979) advocated the need for further experiments then, depending on whether these were successful, a rapid progression towards the provision of two initial digital topographic data bases: one derived from basic scales mapping and the other from the small scales mapping. While the mechanics of creating facsimilies of existing basic scale maps in digital form by line-following means are now well-established (see chapter two and

Figure 1.1a: Complete OS digital map
plotted from a digital tape with the
customer plotting program 'D09'.

Crown Copyright Reserved.



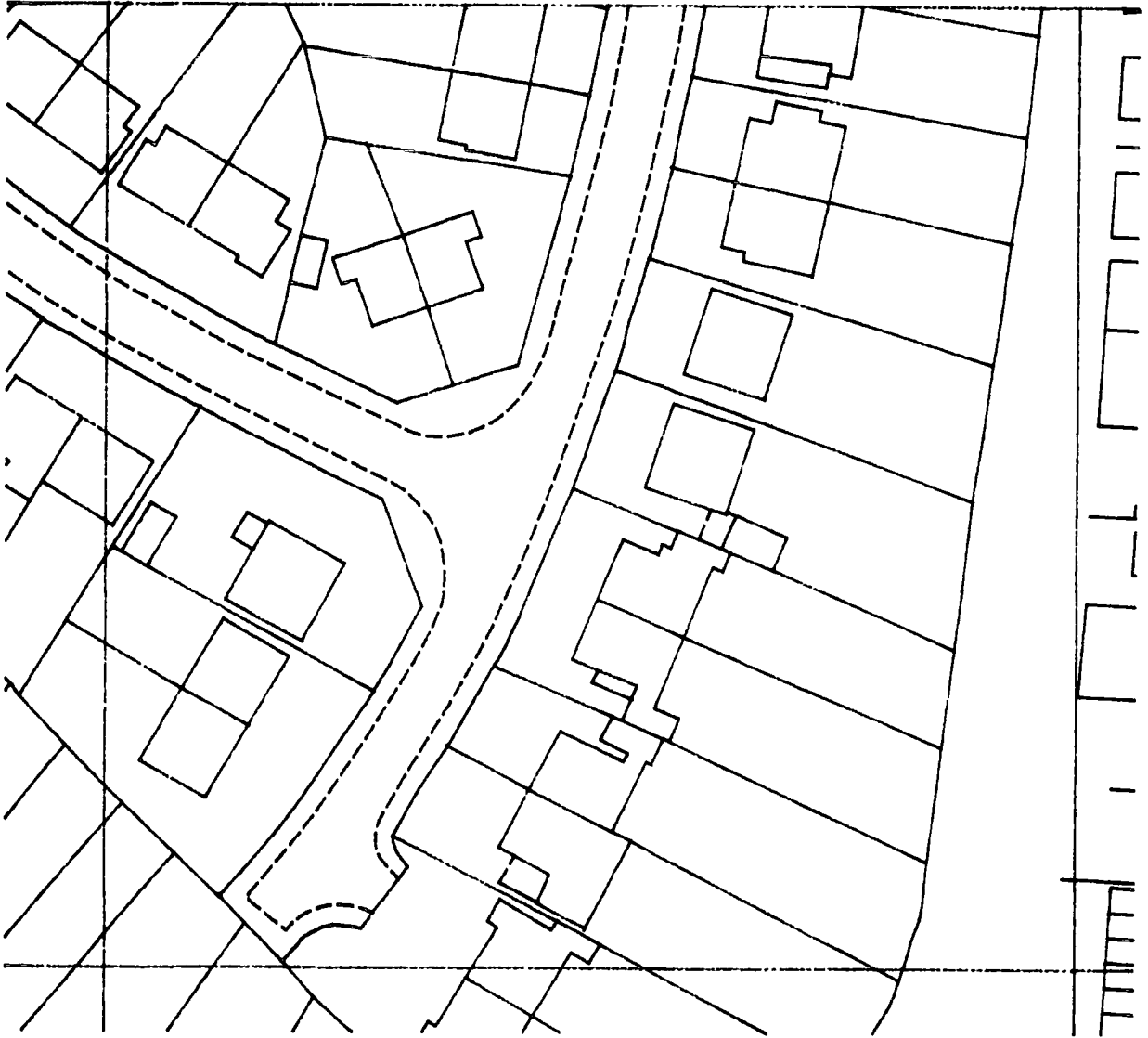


Figure 1.1b: Enlarged section of OS
digital map plotted from the same
source data as those used in fig.1.1a
Crown Copyright Reserved.

figures 1.1a,b for examples of the results possible using OS digital data and plotting software), many fundamental problems of a research nature remain in the collection, up-dating and use of topographic data in digital form (see Rhind 1981, Thompson 1978). The problems involved are fundamentally of a scientific and computing kind, although the difficulty of defining clearly the positive objectives for a working topographic information system, together with a detailed appraisal of user needs also has much importance.

There are, in principle, many tasks for which OS-type data could be used other than simply making facsimile maps; examples of non-graphic uses have been given elsewhere (see Thompson 1978, Dale and Hollwey 1977). Almost all such applications necessitate complete coverage of the country and fast and random access to the data. Existing OS procedures to convert the map data to digital form are proving to be costly and slow (Thompson 1978); the sheer volume and complexity of the data involved in the basic scales (see Adams and Rhind 1981) are not well-suited to manual line-following digitising (see Chapter two). A raster scanning strategy for digital data capture is seen as offering serious alternatives, providing a fast and fully automated method of generating a digital facsimile of a paper map sheet.

1.5 THE OBJECTIVES OF THIS THESIS

This thesis, therefore, examines the potential benefits of a raster alternative to the vector notation and technology used and anticipated for the British national topographic digital data base. Although enormous investment, producing considerable progress, has already gone into raster digitising, (especially in the US military), comparatively little investigation has been made, until hitherto, on the likely characteristics and suitability of such a strategy for OS work methods. In addition to an appraisal of the quality and suitability of raster-scanned data for serious mapping, emphasis is also given in later chapters to the likely characteristics of these grid-based formats for the large scales topographic data, with consideration of the hardware and software resources necessary to produce a workable raster system.

2 VECTOR DATA STRUCTURES

2.1 INTRODUCTION

In chapter one it was shown how the expectations and needs of users requiring topographic information have become more detailed and highly specialised in recent years. This, together with the volumes and complexity of the data involved, has necessitated a move towards the use of computer techniques for the handling of topographic information (Gardiner-Hill 1972, Thompson 1978, McEwen 1980). It was also shown in chapter one how the procedures for the capture and management of the map data within a computer system can be sub-divided into two types under the headings of vector and raster notation.

Raster-based data structures will be considered in chapter three. This chapter will examine vector methods of cartographic data capture leading to vector-based storage and manipulation.

2.1.1 The view of the United Kingdom

Comparatively little work has been undertaken in the UK which involves a raster-based organisation of topographic information. By far the majority of all automated cartographic projects in Britain are based on the vector properties and characteristics of spatial data. By their very nature, computer algorithms are designed and written to solve problems in a logical and step-wise manner. Consequently, much computer software written for tasks (not necessarily restricted to cartographic applications) attempts to automate a system by replicating the stages of the manual approach. This notion is probably one of the fundamental reasons for the comparatively widespread abundance of vector-based data structures for cartographic detail.

The procedures of digital cartography still in use today are not new; it was in the middle sixties when work of this nature began to take shape. The work of the Experimental Cartography Unit (ECU) has made a major impact upon the British and world cartographic fraternity since its formation in 1967 (Rhind and Adams 1980). Although very little published work has emanated from the unit, it has been a world leader in the automation of the map-making process using vector methods.

2.2 SPATIAL DATA

All data contained within a map sheet have specific spatial properties (Tomlinson 1972); they can be conveniently classified into four categories as being POINT (non-dimensional), LINE (one-dimensional), AREA (two-dimensional), and VOLUME (three-dimensional). Any geographical variable can be represented in at least one of the elements of this schema: a location (a point), a boundary (a line), a forest (an area), or an air mass (a volume) are all legitimate. However, in digital terms, the schema need not be as complex as this. Essentially, any volume comprises an area with one added dimension. Similarly an area can be represented by a closed line (a polygon) and, generally, line features can be seen as a collection of connected point locations. In essence, a map sheet can be represented in digital terms simply by a series of points and lines. Although the usefulness of such a data file is extremely limited (McEwen 1980), these concepts comprise the fundamental properties of a vector data structure. Since a map is a series of points and lines, then most computer algorithms have tended to preserve these basic entities in their design.

2.2.1 The incorporation of attributes

Baxter (1976, p127) distinguished between the three components: entities, attributes and time. The last was considered to some extent in chapter one, but certainly the distinction between entities and attributes has importance here. Baxter defined an entity as a 'uniquely identified observation' (points and lines fall into this category). The characteristics with which we choose to describe an entity are termed attributes. Thus by attaching some form of attribute or description to the basic entities it becomes possible to represent and describe the component parts of a graphic medium (map sheet) in a digital format. In summary, a single point location can be represented in three dimensional space by four parameters: X, Y, Z and W. The first three fix the position of the point in geographical space and the W term can be some singular or string of attributes or feature codes from a predefined classification. W is different in character from X, Y, Z in schemes which incorporate multi-feature coding; in such instances W itself will be a vector quantity.

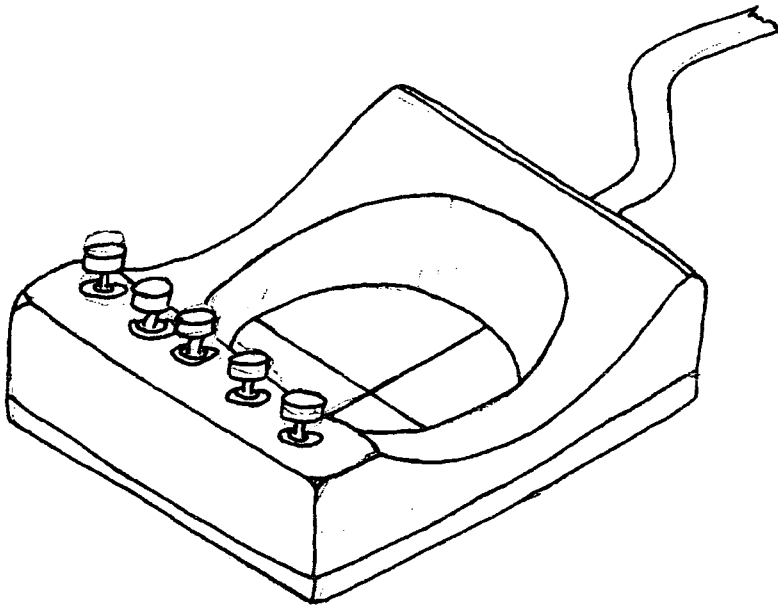
2.2.2 Data capture

Digital encoding of topographic information can be carried out by various means, the sophistication of which is usually dependant on the volume of data to capture. The most simple and straight forward approach could involve the measurement and coding of National Grid coordinates for subsequent key-punching on to cards. Clearly such an approach could be only considered in very trivial projects but nevertheless could be cost effective in the right situation. Rhind (1974) suggests empirical evidence indicating it to be faster and cheaper to utilise this method where jobs occur spasmodically and are limited to approximately five hundred point values or less.

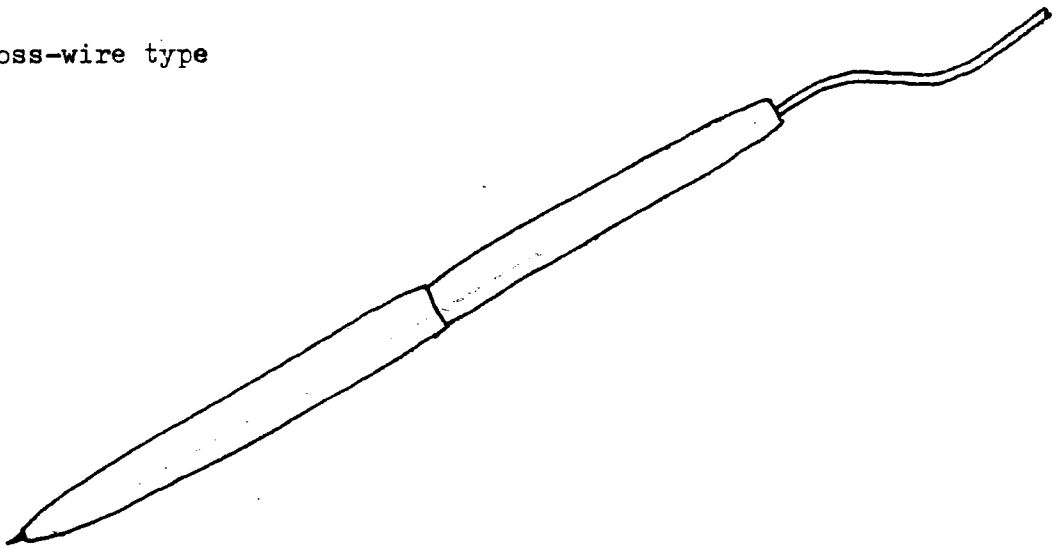
Most digitising projects today, however, utilise specific hardware devices designed for undertaking the graphic-to-digital conversion with relative ease. Vector-based digitisers have traditionally been manually controlled by trained operators, although much more recently the technological advances in lasers and solid state circuitary have seen the introduction of semi-automatic instruments. These will be considered later, although first an examination of manually operated devices will be given since they have been commonplace in most cartographic data

encoding for over a decade and a half. Rogers and Dawson (1979) give an overview of hardware technology for manual digitising. Essentially, the hardware design of digitisers has changed little over the last decade when compared with the vast increases in computer power for the same cost. Manual digitising is still predominantly carried out on the same type of device that was in use in the late sixties albeit with more 'intelligence' offered with the advent of micro-processor technology. Consequently, digitising procedures and interfacing to computers have been much improved, producing vastly more efficient systems.

The principles behind manual digitisers are simple in concept. The position on the table-top of a moveable cursor is sensed by a mesh of wires embedded within the table. Digitising tables are available in a wide variety of sizes ranging from as small as 25cm x 25cm to approximately 1.5m x 1.0m; the larger models have more direct uses in cartographic applications because they can accept the largest of map sheets. There are two types of cursor (see figure 2.1) in common use today: the stylus which has an appearance similar to a normal ball-point pen and a crosswire cursor providing for more precise pointing since the body of the cursor does not obscure the line being digitised. Although the crosswire type of cursor is more suited to cartographic work, the stylus type has much potential in the areas of computer



(a) cross-wire type



(b) stylus type

Figure 2.1: types of moveable cursor on manual digitising tables

graphics which require more artistic flair. Recently graphics designers at the British Broadcasting Corporation (BBC) have been producing extremely clear artistic effects for television graphics using only small digitising tablets and a stylus type cursor (BBC pers comm 1980).

Manufacturers usually provide some form of interface with their devices in order to connect them to data storage facilities, and these interfaces are now normally centred around a micro-processor control which can perform scaling and transformation of table coordinates to the user's reference system. Until the early seventies, it was commonplace to treat digitisers as off-line peripherals to computers; attachment to a magnetic tape drive was thought to be far more cost effective than a direct link to the central processor. As an off-line process generally, it is possible to link a digitiser to any of the following items of equipment: floppy disk drive, paper tape punch, card punch, video display device, or magnetic tape drive. The advent of far more widespread interactive computing, together with the dramatic cost reductions of hardware and solid state circuitry in recent years, has made the idea of on-line digitisers a much more worthwhile proposition. The practicalities and potential of interactive data editing could only have been realised with digitisers being connected directly to an automated data processor. With such an

installation, instantaneous responses are obtained from keyboard or digitiser instructions providing real-time data validation and manipulation.

In general terms, there are two ways of capturing data in digital form from a digitiser under human control. Firstly it is possible to digitise individual points at the discretion of the operator. This is POINT DIGITISING and is usually slower and more laborious than the second approach which is STREAM DIGITISING. In this second mode of operation, the device will continue to supply coordinates whilst the cursor is touching (or in close proximity to) the digitising surface of the table. In stream mode more coordinate pairs are generated for each feature since there no longer exists the selection capability which an operator can have in point mode. However, on most modern devices it is possible to set the flow rate of the coordinate stream from, say, one point to around one hundred points per second (Summagraphics Corporation Digitiser User Manual 1979). A decision of the number of points required to represent linear entities is arguable; most design schemes have their own 'rules of thumb', although generally more points are required for curve data whilst straight lines can comprise much lower frequencies (figure 2 in Adams and Rhind 1981). Rhind (1974) has pointed out how such considerations should be predefined to ensure data uniformity.

Micro-processors have made the most substantial impact on these hardware devices in recent years, providing a more flexible operation and ease of use. Some have the ability to suppress the supply of coordinates until they differ by a preset tolerance from the previous pair sent, thereby reducing the chance of point repetition and hence redundant data. There are many models on the market today with varying characteristics and specifications (see Rogers and Dawson 1979). Resolution capabilities seem to range from 0.0025 to 0.5 millimetres; for most cartographic applications an acceptable value would be in the range 0.05 to 0.25 mm. In quoting the required resolution of the tables for cartographic work it is equally important to include a mention of the acceptable accuracy of any feature positioning; this can be no better than the resolution and will require to be in the range 0.1 to 0.5 mm.

2.3 DIGITISING PROCEDURES

There are many variations in present day encoding procedures and some examples of the major world mapping systems - using vector digital methods - will be considered later in this chapter. Generally, only subtle differences in work methods have any importance; the basic underlying operations remain the same in any data collection system.

The setting-up procedure is similar in most digitising agencies: four or more control points of known position are carefully digitised to enable such procedures as the scaling and transformation of local table coordinates into a more useful coordinate system. When the suitable transformation parameters have been computed, it is possible for features on the source document to be digitised in any desired order; some systems, however, require digitising to be carried out within a formal set of rules.

The inclusion of feature codes with the digitised points and lines is achieved by a number of methods, although almost all of them involve data-tagging at the same time as digitising is taking place. The simplest technique from a programming point of view involves typing into an adjacent keypunch the feature types as they are being encoded. Although slightly different in concept (the system uses a photogrammetric form of data capture), the digital mapping system in use at Huntings Surveys Ltd. (Keir 1976) provides a good example of such a system. This approach can seriously affect work throughput since the table operator has to repeatedly move between the keyboard and the moving cursor. A more useful technique is a facility whereby attributes can be entered from the digitising cursor. Some cursors have a number of buttons to facilitate this; certainly they can be used for including complementary information such as start

and end of lines, curves, polygons, etc. However, it is the use of menu techniques which has been the most commonplace method of assigning feature types on manual tables in recent years (see Coles 1974). The menu method involves allocating an area of the digitising table for attribute assignment. The area is sub-divided into a series of small square units; the position of each square is programmed to represent some feature type. Digitising a point within the desired square in the menu area instructs the computer to assign a given attribute to the feature which is to follow. The Ordnance Survey has used menu techniques very successfully for the last decade in their digital data flowline (Coles 1974).

2.3.1 Data validation

The distinction between data capture being either off-line or on-line has been instrumental in the way that data validation has developed in the last five years. Most digitised data sets contain errors, especially those data which have emanated from an off-line table and are stored immediately. It is the aim of the editing stage to detect and eliminate these errors, or reduce them to some acceptable level. In many instances, the level of tolerable error should be higher (and hence worse) than is at first thought

necessary since it is common to find digital data being used at some later date for purposes which they were not originally intended. Since the middle 1960's, off-line data editing has been the primary data correction technique, although more recently an on-line interactive approach has been very successful in improving efficiency and reducing costs. Today error detection may be carried out in part by machine and in part by humans; inconsistencies within the data such as an absence of a feature code or an incorrect tally of start and end of feature markers can be detected by the computer in some post-hoc process. These are brought to the attention of an operator who will be able to take some appropriate action. Complete omissions or errors which are consistent within the schema of the system's specification cannot be detected by machine and require manual validation. Such checking is tedious and error-prone; the use of off-line edit plots and visual scans through coordinate listings results in a slow turn-around with many inefficiencies. This approach has not been cost-effective for large data sets (Thompson 1978). The ECU implemented an alternative approach in the early 1970's to attempt to alleviate the inefficiencies of off-line verification to some extent. This involved the automatic checking of double digitised data. Essentially, ^{though,} it was found not to be cost effective (Rhind 1974) to digitise the data twice (by two different operators) and then superimpose the data sets within the

computer to search for inconsistencies. The procedure was dropped, therefore, soon after its initiation.

Once errors have been detected by procedures such as those outlined above, it becomes necessary to make corrections to the data file. Some corrections may need a return to the digitising table, others can be handled off-line as a batch job of instructions. For example, such inconsistencies as missing feature codes or the mismatched end points of polygonal data can be corrected off-line, but the insertion or amendment of whole features require a return to the table. The costs incurred by data checking and edit stages are alarmingly high in most systems (Thompson 1978).

It is here where the the efficiencies of interactive graphics are proving to be most cost effective (Allen and Morris 1980). Whole maps or selected portions can be displayed very quickly on storage cathode ray tubes. Operators can reference particular features in real-time by cross-hairs on the screen which are controlled by some form of joystick on the terminal panel. Once a feature has been identified, software can search through the data file and perform most correction functions on it: deletion, amendment, redefinition of feature type are all possible. These systems require highly sophisticated software and data structures. This, together with the high cost of graphics terminals has reduced their widespread availability within the topographic

mapping community although this situation is beginning to change with the advent of computer systems designed for map data. The Ferranti-Cetec CLUMIS system (Ferranti-Cetec Ltd., 1979) and Laser-scan Laboratories IGES system (Laser-scan Laboratories Ltd., 1981) are two British examples and there are numerous systems emanating from the United States such as those from M and S Inc., Broomall Industries Inc.

2.4 DATA ORGANISATION AND STORAGE

Appropriate formatting and organisation of data for storage within a computer system has great importance in most digital mapping systems. At the data capture stage, for instance, it is necessary to structure the information in such a way so as to facilitate easy verification and edit tasks. However, this is not to suggest that a data bank of topographic information should be structured purely in a manner which suits the data capture stage. Once a data file has reached an adequate standard within the limits of the system's thresholds, it can and should be reorganised in such a fashion as to facilitate efficient storage and retrieval methods. Many digital systems incorporate two different data structures for data capture and data banking purposes; Ordnance Survey has used such an approach in its digital system.

An efficient structure for vector-based cartographic data has been a subject of much theoretical debate. Data base organisation is an area of the computing industry which is now beginning to reach a high level of software expertise; the work of pioneers such as Edward Codd and James Martin has been very influential. However, vector-based cartographic data do not fit neatly into the relational and Codasy1 ideas of general data base systems (Martin 1976), although Cox et al (1981) suggest a possible solution.

Data base software is no longer a new concept; many nationwide organisations, such as those connected with banking, insurance and travel are incorporating large and efficient computerised data stores into their business as a rudimentary part of the system. Although their data have somewhat different characteristics to those of topographic information, some lessons can be learnt from these data base management system (DBMS) pioneers. Indeed, the United States Geological Survey (USGS pers comm 1980) are making use of a propriety commercial data base system for interim operations.

2.4.1 A suitable data structure

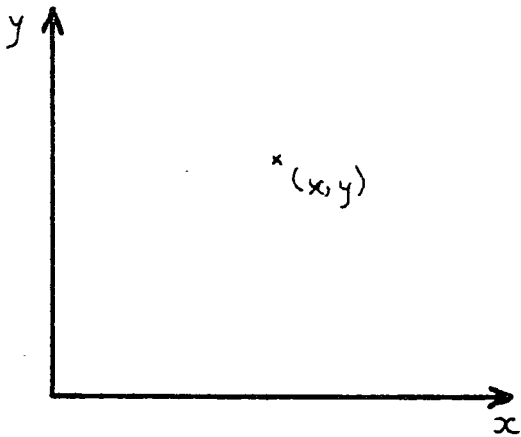
When designing a suitable structure and method of organisation for vector-based coordinate data strings, it is important to consider the potential needs of users at the outset. Such an approach can expedite easy and efficient retrieval operations in later stages. Chapter one has attempted to categorise the types of user requests expected to be made to a topographic information store; in their simplest form, typical requests will involve the recall of specific areal units or polygons from a complex network of data for the whole country. Polygonal data sets of this type abound and almost all use vector methods of structure (DoE 1976, US Bureau of the Census, pers comm 1980). Requests for polygonal data are extremely common for areal-based analyses by social scientists and planners involved in population census statistics (Rhind et al 1977), land use (Bickmore and Molineux 1978, Rhind and Hudson 1980) and climatology and meteorological information (Harris and Barrett 1978).

The representation of areal data by vector strings needs careful planning and structure. Numerous suggestions for the storage of areas have been present though, in essence, vector strings are only able to represent the boundary between areal units. Some data structures handle the area itself as the

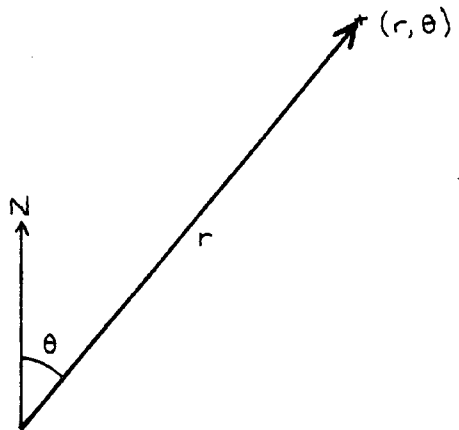
basic spatial units (BSU) although in all cases these are where the BSU's are of predefined constant size and shape. Grid square census data are a typical example (Rhind et al 1977, Soil Survey of England and Wales pers comm 1981). In instances where the boundary information is the BSU then all area manipulations must be derived from these by careful selection and aggregation of the desired vector segments (DoE 1976).

Data structures representing areal units specified by the boundary positions require complex methods of data tagging. The most common technique in current use is the addition of two feature codes, a left and a right attribute, for each boundary (DoE 1976, SDD 1978). In order to preserve a convention at the data capture stage, digitising operators are instructed to encode the polygon boundaries in a continuous rotation - clockwise or anticlockwise (Loomis 1965). Leaving aside techniques where areal information can be referenced by single points (see Pfaltz and Rosenfeld 1967) or by grid squares (see Rhind 1981), all polygons are normally represented conceptually as strings of straight line segments though a few systems have treated polygons as assemblages of parameterised curves, such as arcs of circles. In general, such an approach can only approximate the real world situation and the accuracy is a function of the length of the segments chosen with respect to the degree of

convolution of the polygon (DoE 1976, Adams and Rhind 1981). The position of each vector quantity is held as a series of absolute rectangular cartesian coordinates (providing the difference between the end points of the segments in two mutually right angle directions), or exceptionally as polar coordinates (providing an absolute position, followed by a vector radial quantity and orientation).



Rectangular Cartesian



Polar

There have been variants on these basic themes such as those outlined in Freeman (1961) and Tomlinson (1972) which can, in certain situations, have distinct advantages, though Baxter (1976 p 137) has pointed out several limitations with these techniques. The storage of polygon boundaries as closed cartesian coordinate strings is well suited to the data capture stage although distinct disadvantages do exist without appropriate data structures. Although the process involves extra digitising, generally there need be fewer attributes for the whole structure. With such a method each boundary has to be held twice - once for each areal unit. Apart from increasing the storage requirements virtually by a factor of two, the duplicated boundaries are unlikely to be coincident - a factor commonly associated with manual digitising. A more amenable approach outlined by Loomis (1965) and demonstrated by Baxter (DoE 1976) is to represent boundaries as sections; each section comprising the line segments between two adjacent areas together with the associated left and right feature code. Baxter (DoE 1976) has demonstrated a hierarchical method of feature coding for polygonal data for a parsed data set of British Local Authority Boundaries. At the most disaggregated spatial level are the Metropolitan and Shire districts in England and Wales and Regional districts in Scotland. Superimposed upon these units are the standard economic planning regions for Britain. Each district on the base map has a unique code

assigned to it. These codes are constructed hierarchically in order to provide additional information about the parent region. Thus a six digit code which represents the basic spatial unit has its two most significant digits representing the planning region, the middle two digits representing the county and the two least significant digits designating the district within that country. Examples of this coding scheme are given in table 2.2.

Table 2.2 (after DoE 1976)

Codes for districts within counties within regions

Region	County	District	
East Anglia	01 Cambridgeshire	0101 Cambridge	
		010101 E Cambridge	
		010102 Fenland	
		010103 Huntingdon	
		010104 Peterborough	
		010105 S Cambridge	
	Norfolk	0102	010201 Brockland
			010202 Broadland
			010203 Gt. Yarmouth
			010204 N Norfolk
			010205 Norwich
			010206 S Norfolk
			010207 W Norfolk
	Suffolk	0103	010301 Babergh
			010302 Forest Heath
			010303 Ipswich
			010304 Mid Suffolk
			010305 St Edmundsbury
			010306 Suffolk Coast
			010307 Waveney

An alternative method for holding areal data has been to approximate the polygon boundaries by mathematical functions. The use of such techniques (high order polynomials or Fourier transformations) tends to reduce the number of coordinate strings although the overheads required for the accumulation of the function coefficients have a sizeable demand for computer storage and processing mill-time. Many vector-based systems compromise between the use of many coordinate strings and the polynomial techniques. The Ordnance Survey approach, for example, has been to store straight line data as coordinate strings at the capture stage and use splining techniques to generate the curves. The digitising operator has merely to digitise three or four points for each curve and further points are generated post-hoc along the spline fitted to these points. Even so, the Ordnance Survey has found it to be more advantageous to store the generated coordinate strings in their data bank as opposed to holding the polynomial functions.

2.4.2 Data characteristics

There is increasing evidence in conference reports and the technical press to suggest that many organisations and individuals throughout the world are either creating or using

coordinate data bases. Although few inventories exist, a recent compilation has been attempted at the University of Durham for the International Federation of Data Organisations (IFDO). From this and other evidence, it is known that numerous individuals have separately digitised coordinate data sets for their own purposes from identical source documents. Generally, such users tend to treat this topographic or cartographic information as a geometric framework on which they can 'hang' their own specific attributes - typical examples in the UK are the public utilities: telecommunications, electricity and gas, all of which superimpose their information on to Ordnance Survey base data (Gardiner-Hill 1972).

It seems increasingly evident (Rhind and Adams 1981) that many data base managers and certainly the majority of the geographic community pay little attention to the statistical characteristics and properties of the base data with which they are concerned. Little work of this nature has been reported in technical journals and conference proceedings. Clearly these considerations are not an academic pastime; the design or selection of a suitable development programme and satisfactory computer system becomes impossible without some specification of task and of data characteristics. The storage media, the retrieval performance, the methods of revision and the design of any processing software are all

dependant on the nature of the data and the volumes involved. The United States Geological Survey (USGS) is generating 70,000 megabits of information annually from their digital elevation model data alone. This data volume factor will be a crucial problem in the development of their cartographic data base (DCDB) system (McEwen 1980). Even with the use of high density 6250 bpi magnetic tapes, it will take at least one hundred storage volumes every year just to hold this information (without backups) prior to any post-hoc processing which may take place. Such vast volumes of data require an effective data base management system to expedite a controlled development and operational programme for the USGS.

The first attempt to categorise and project the size of a digital topographic data bank in the UK was undertaken by Howard (1968). Her study was based on manual measurements of line length and point counts of a necessarily small sample (0.5 per cent of area) of the 'one-inch' series - now superceded by the 1:50000 Landranger series. Although her somewhat generous assumption of digitised point frequencies (8 points per millimetre) do not concur with current OS digitising conventions, her results are still of interest: 555 million coordinates were anticipated to be necessary to represent the 1:63360 scale map data in digital form. She suggested that the total line length on the one-inch maps was

of the order of 70 kilometres and, of this, about half was due to contours. A more recent comparative study (Haywood pers comm 1980) for the 1:50000 scale mapping series has suggested figures of 57 km and 60 per cent respectively for these measures, although this later project was based upon different methodology and involved specific digitising tasks.

Similar work has recently been carried out to determine the characteristics for the Ordnance Survey basic scales topographic data in Britain. Mapping at 1:1250 and 1:2500 scales comprise the bulk of the workload for the National Mapping Agency: this factor, allied with suggestions (HMSO 1979) that complete national digital coverage of these large scales data should be made available as soon as possible, makes a knowledge of such characteristics highly desirable.

A recent study (Adams 1979, Adams and Rhind 1981) has quantified the nature and content of a sample of OS map sheets which has led to a prediction of the likely size and characteristics of a digital topographic data base for the whole of Britain. The study was based upon the statistical analysis of 478 map sheets selected as a stratified sample from the 7000 in digital form at the end of 1978. The projections are based upon a number of important assumptions which are itemised in Adams and Rhind (1981). In summary, however, the results are outlined below:

Chapter two

1. In terms of data volume, the 1:1250 urban sheets are more dense than the 1:2500 rural sheets by between two and five times: the 1:1250 data contain far more linework than do the 1:2500 data (on average 35m and 12m respectively).
2. Despite this, the main component of the data bank will be the 1:2500 scale maps since there are three times as many of them (164,461) as the 1:1250 maps.
3. Apart from the scale stratum, the efficacy of the a priori stratification was shown to be dubious - at this scale of map, the data content appears not to be predictable from intuitive notions of topographic type or population. Later unpublished correlation studies by the author with topographic data and census of population data have shown this to be true.
4. The vast bulk of the data is linework and many of the frequency distributions of points within lines and some point spacing are markedly skewed. (See tables 2 and 3 in Adams and Rhind 1981). For this reason, mean values should be treated with caution.
5. No contour data are present in the 1:1250 and 1:2500 scale maps: on this basis, and accepting all the

limitations of the 0.1 per cent areal sample, projections for the total data base were made (see tables 5 and 6 in Adams and Rhind 1981). If updating were to follow standard manual practice it would be on a continuous basis, presumably sent from regional offices; much of the principal data ought to be on-line or at least loadable within a short time.

6. Very considerable amounts of data compaction are possible using a battery of techniques. Certainly in-house data structures at the Ordnance Survey can achieve a factor of five times reduction in storage requirements over formats released to customers.

Although there have been other smaller studies of large scale topographic data bases, for example in the Survey Department of Public Works of The Netherlands (Tienstra and van der Kraan 1975), no other studies of a similarly detailed nature are known to the author. In conclusion, it is anticipated that the total basic scales data at 1:1250 and 1:2500 will comprise 3200 kilometres of mapped line length being represented by 1200 million coordinate points. In essence, using current customer export data formats will require of the order of 580 computer compatible magnetic tapes (standard 2400 feet, 9 track, 1600 bpi) to store one

copy of the national large scales archive. Even so, the data volumes projected are no larger than those held by other agencies such as bankers or airlines and accessed on a daily routine basis. New developments in mass storage modules are bound to be commonplace within the next five years. There is every reason to believe, therefore, that the storage and retrieval of a national digital topographic data base is a feasible operation within the bounds of existing technical and management knowledge.

2.5 THE DIGITAL MAPPING APPROACH OF ORDNANCE SURVEY

Great Britain is by far the most comprehensively mapped country in the world, and the Ordnance Survey is internationally recognised as one of the best organisations of its kind (Harley 1975). Following the recommendations of the Davidson Committee of 1938, the terms of reference under which the OS has operated have been to produce and maintain basic surveys at 1:1250 scale for major urban areas and at 1:2500 or, in some instances, at 1:10,000 scale for the remainder of the country (Harley 1975, HMSO 1979, Seymour 1980).

Around a decade ago and following intense stimulation from organisations like the ECU, the National Mapping Agency

inaugurated a digital mapping pilot scheme. The objectives at that time were seen as the evaluation of automated methods from a technical and economic point of view and to stimulate user interest and demand (Gardiner-Hill 1972). Although a visual product for the topographic data was likely to remain a major requirement for the foreseeable future (maps produced by digital means were to be made indistinguishable from conventionally created maps), it had become obvious that demand was growing for the data to be made available in machine-readable form. From the Ordnance Survey point of view, holding the data bank in digital form was seen to offer the following benefits:

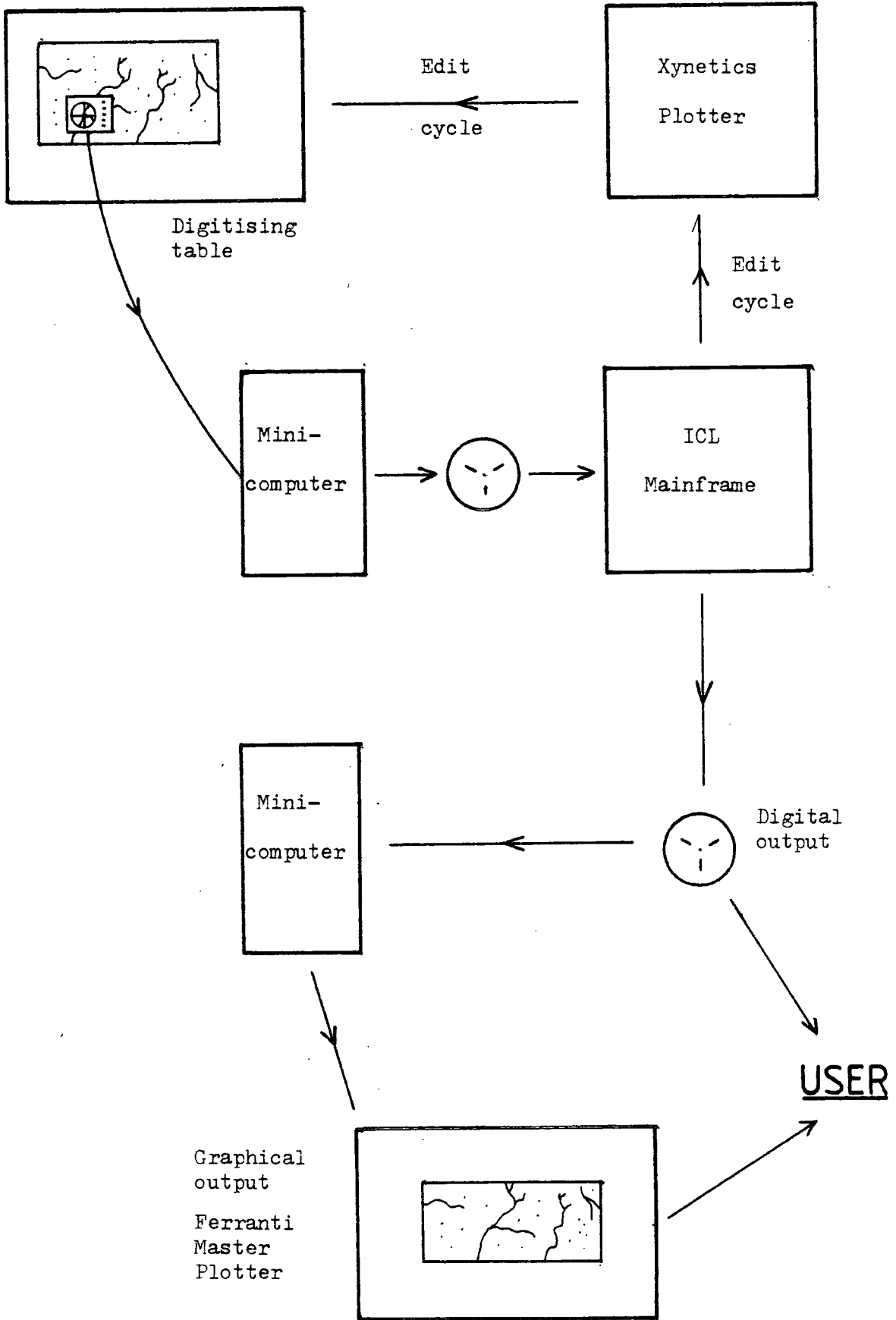
1. Savings of space and costs: the handling of over 200,000 paper sheets is a daunting task, although attempts had already been made to alleviate these problems with the use of microfilm techniques (Marles 1971, Adams et al 1980).
2. Security of the survey record: large scale map detail would no longer have to be stored on dimensionally stable materials. Coordinates held in computer files can always be recalled at their original accuracy.
3. Greater flexibility of usage: customers could merge the topographic base data files with their own specialist

information far more quickly and easily.

4. Ease of derived mapping scales: at the time it was thought to be, at least, conceptually possible to modify the basic scales to derive smaller scale mapping. More recent experimentation has shown the derivation of 1:25,000 mapping to be difficult because of the problems of selectivity and generalisation. However, quite successful 1:10,000 mapping has been possible with the need for only limited additional effort by draughtsmen (Searle and Waters 1979).

The digital mapping production flowline has remained virtually unchanged since its inception, and is based on what can now be considered conventional manual digitising procedures: batch processing of data and the production of edit plots from which the necessary corrections are made by further digitising and off-line instructions to the computer. (See figure 2.3). The system can be classified into five stages and these are discussed below.

Figure 2.3: Description of the Ordnance Survey (DMB) large scales digital flowline



2.5.1 Preparation stage

The source documents used are the master survey documents (MSD) - the dimensionally stable surveyors' working drawings. In order to achieve the greatest possible accuracy at the digitising stage, the 1:1250 and 1:2500 MSD's which are held at the correct scale, are photographically converted to negative form and enlarged by a ratio of 3:5. Thus the 1:1250 data are digitised at 1:750 and the 1:2500 data are digitised at 1:1500 (thereby increasing the linework to be digitised by a factor of 1.6). In addition, two diazo copies of the graphic are made: one of these is to be the progress-diazo which is an aid to the digitising operator in monitoring the progress of the data capture. Fibre tip pens are used to draw over the information as it is encoded; a different colour to represent each day's work. The second diazo copy becomes the pre-edit plot on which all text data are serialised and their start positions marked. Textual information including the desired text size and font is entered onto specially prepared coding forms in serial order for subsequent key-punching into the computer via a processor controlled key-punch (PCK) system which comprises, essentially, eight video terminals connected to a controlling mini-computer which interleaves all input data on to a magnetic tape drive for transportation to the mainframe computer.

2.5.2 Digitising stage

The Ordnance Survey has twenty four manual Freescan digitising tables manufactured by Ferranti-Cetec Ltd. Nine of the tables output coordinates directly on to their own magnetic tape drives and the remaining fifteen are connected to a processor controlled digitising (PCD) system which is based on a Digital Equipment Corporation PDP 11/05 mini-computer. This computer system interleaves the output from all 15 tables on to one magnetic tape which can be sorted subsequently by the departmental mainframe computer at the end of the day (Harris 1980). In essence, a PCD-based system has the distinct advantage of compacting the output of a day's work on to one magnetic volume but is highly dependent on the reliability of the 11/05. Should the processor break down, then all fifteen Freescans are out of commission; a three hour breakdown thus implies forty five hours of lost digitising time. Fortunately, the 11/05 at OS has been extremely reliable (Linklater pers comm 1979). The digitising procedures that have been adopted over the last decade generally follow normal conventions with predefined specifications: a thirty-six point set-up process is performed for 1:1250 map sheets (nine points for 1:2500). Point digitising methods are used with a data menu for feature code assignment (see section 2.3, Coles 1974). Curves are encoded as a minimum of three points and splining

methods generate further points as a post-hoc process. The squaring of buildings is also carried out as a post-digitising basis.

When the progress-diazo indicates the map sheet to have been completed, the digital version of the data can enter the edit loop of the system. The time taken to complete a map sheet at the digitising stage is a function of the data complexity and line lengths involved. Personal research by the author during secondment to Ordnance Survey headquarters in 1979 has shown digitising times to range from three hours (sheet TQ 6693SW) with a line length of 899 cm to around forty hours (sheet SP 0689NE), with a line length of 6192 cm. Taking the total times (preparation + digitising + edit + text collation) for a selection of the 478 map sheets used in the Adams (1979) study, a least squares fit was determined for a general mathematical expression to link digitising times with map line length and feature frequency. A description of the derivation is given in Appendix 2.1. The expressions derived were as follows:

(a) 1:1250 scale:

$$T = \frac{L}{1.75} + \frac{N}{144.1}$$

(b) 1:2500 scale:

$$T = \frac{L}{1.10} + \frac{N}{128.2}$$

where: T = total digitising time in hours
L = line length in metres at plot scale
N = number of features per map sheet.

2.5.3 Edit stage

The edit-loop process in the production flowline is inevitably slow: all verification and correction is made by off-line processes. The first edit plot is produced in a series of stages. The departmental mainframe computer - an ICL 1906S - spools a plot descriptor file (PDF) on to a magnetic tape whence it can be taken to one of two Hewlett Packard mini-computers which drive each of two Xynetics model 1050/1150 plotters. In essence, a PDF is a file of binary information which is subsequently interpreted by appropriate software to drive specific graphics hardware, such as hardcopy plotters or storage tube terminal devices; it includes all coordinate information, controls on the colour of ink required, the intensity of the plotted line, the size

of plot and so on. The Xynetics Series 1000 are high speed flat-bed plotters capable of speeds of up to 40 inches per second.

The flow-line edit plots are multi-coloured to aid data verification: blue ink represents all buildings, fences are shown in black, road casings are represented by green lines and all minor buildings and other data to be suppressed in derived smaller scales are shown in red. These plots are useful for checking the consistency of the data, their accuracy and appearance; their multi-colour nature facilitates a verification that features have been coded correctly.

Corrections need to be made to the digital files as they are detected on the edit plot; some may require a return to the digitising table, whilst others can be performed as a batch job of computer instructions. This edit cycle is repeated continuously until the digitising draughtsmen consider the data file to be acceptable. On average 3 - 4 edit plots are produced per sheet which takes approximately 20 minutes to plot on the Xynetics plotters (Smith pers comm 1979) though plotting time is dependant on data volume.

2.5.4 Examination and archiving stage

As soon as a digital file has passed scrutiny at the edit stage, a master-plot is produced. Ordnance Survey operate two high quality Ferranti Master flatbed plotters which are controlled in a similar way to the Xynetics plotters - 'off-line' to the mainframe under the control of a PDP 11 mini-computer; they operate at much slower speeds than the Xynetics plotters (typically 4 inches per second) and plot directly on to film. These plots are drawn at correct publication scale; this is the first time this will have occurred since the enlargement process, which was carried out photographically at the preparation stage. In general terms, the turn-around times of the edit stage are not acceptable for a production system. The turn-around times of the various stages for 1:1250 data of the Birmingham area - which are inclined to be voluminous (Adams and Rhind 1981) - are shown below:

Digitising	--	3 - 4	days
Xynetics edit plot request	--	8 - 10	days
Master plot request	--	4 - 6	weeks

(Source OS pers comm 1980).

When a digital sheet has reached the master plot stage, it passes from the Digitising Branch of the Cartography Division to a special Examinations Branch. This branch checks all 1:1250 and 1:2500 conventional and digital mapping for accuracy, completeness and conformity with the specifications of the day prior to reproduction. At this stage most corrections necessary are made by hand to the film positive although records are kept to ensure that the digital files are updated at some time in the future. When the Examinations Branch has approved the content of the digital files, they can be archived, which involves reformatting the files on disk and transforming them to magnetic tape for storage. The OS system is highly dependant on magnetic tapes: in 1980, the mainframe had twelve tape-drives. Information is only held on-line (magnetic disk) whilst it is in the digitising and edit stages. For security purposes, three copies of the data archive are held: two are stored in different locations within the Southampton headquarters and one is sent to the Office of Population Censuses and Surveys (OPCS) in Titchfield.

2.5.5 Cartographic enhancement stage

There are still a number of enhancements that have to be included before the published plan can be reproduced and sold

in its graphic form: vegetation, parcel area values (at 1:2500 scale) and some marginal information are added by hand and the building infill mask is cut. The three components are combined photographically to produce a film positive for subsequent reproduction by conventional means.

The status quo cannot be considered as providing a satisfactory production flowline: it is inefficient and expensive. The digital flowline requires 92 man hours per sheet compared with 68 man hours for the conventional flowline (Thompson 1978) and hence costs 1.6 times the conventional system. The reasons for these unfavourable comparisons with the conventional processes were outlined in the Review Committee report (HMSO 1979). The report stated (para 10.5) the main reasons appear to be:

- a. "The large number of processes involved. In the present OS system digital work has to be plotted, checked and corrected, on average 2.5 times before the final map can be produced. Fourteen separate operations are involved, compared with eight for the conventional manual production system;
- b. the relatively large number of conventional processes still to be undertaken on each digital map (for example the stipple infill of buildings);
- c. the costs of checking (particularly in the case of feature codes attached to all lines and points)."

With a current throughput of 2000-3000 maps per year (Logan 1981), the estimated timescale to complete the digital conversion alone without any effort being placed on the revision of existing mapping is approximately seventy years. At least a six-fold increase in work rates would be required to complete digitising within a ten-year period as has been proposed in the Serpell Report (HMSO 1979). The achievement of such a goal is almost certainly allied to the adoption of automated digitising techniques, particularly since plans announced in early 1982 envisage a reduction of one hundred cartographic draughtsmen. There are two alternatives available for automation: raster scan based techniques or vector following techniques. Detailed consideration of these approaches will be given later.

2.6 THE NEED FOR TOPOLOGY IN THE ORDNANCE SURVEY DATA STRUCTURE

OS digitising procedures as outlined earlier have been organised in such a way as to provide a tolerable job for the digitising operator; such terms of reference made good sense for the initial objectives of the system - the production of high quality graphics (Gardiner-Hill 1972). It is, for example, unnecessary to have adjacent features cross-referenced if further mapping is the only digital application - the lines may be drawn at different times but

that is irrelevant provided they are drawn in correct relationship to each other. Versatility in the type of graphic to be produced is easily achieved with the present data structure (DMB): customised images with scale, area of interest and feature type can be produced economically and easily.

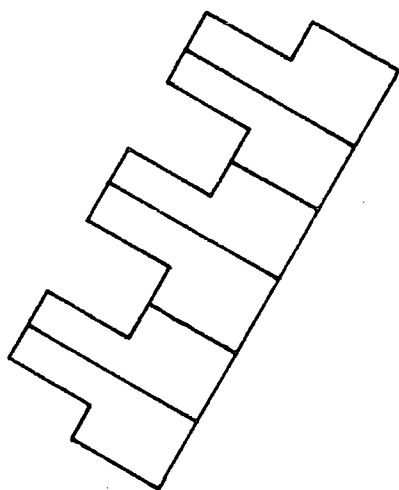
The national mapping agency has been fully aware of growing interest amongst users wishing to use digital topographic data for other purposes. Thompson (1978) suggests some typical examples: gazetteers, rural and urban property data bases, road and transport data bases and public utility networks. The need for easily accessible and carefully structured digital land information has been long argued by Dale (1977; Dale and Hollwey 1978); there is no reason why a topographic data base should not be coordinated with land use studies, pollution monitoring and terrestrial ecology as well as data provided from population censuses; many (over fifty per cent) of the boundaries of the census Enumeration Districts have remained constant between 1971 and 1981 and many are defined along physical features recorded on OS maps. To achieve analytical possibilities such as these is impracticable given the data structure described above. A typical case is the identification and retrieval of land parcel boundaries, another is the lack of any definite linkage between text and hard detail within the data.

2.7 THE RESTRUCTURING PROJECT FOR ORDNANCE SURVEY DIGITAL DATA

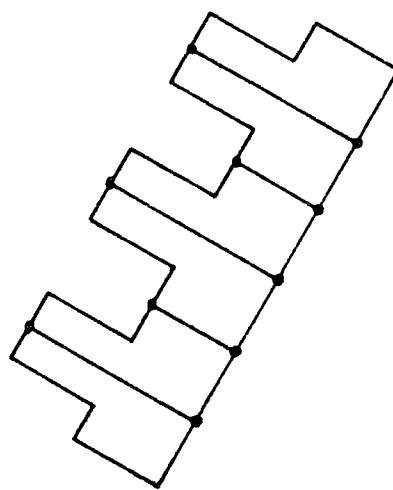
The data structure generated by the existing digital flowline is unsuitable for any of the non-graphic uses discussed in section 2.6. For these reasons, a feasibility study was initiated by the OS to "investigate the different needs of potential users of OS large scale digital map data in local authorities, public undertakings, central Government and elsewhere" (Thompson 1978 p 11). In 1975, PMA Consultants Ltd. (a commercial software house) was commissioned to reformat existing data into a format more suited to analytical processes (Thompson 1978, HMSO 1979).

In concept, restructuring is trivial; the existing data need to be broken down into basic entities and then recombined into units more appropriate to user needs. In general terms, the basic entities of the map data are the collection of x,y points and any restructuring process involves merely reconnecting these points into more appropriate units and regrouping the attributes where necessary.

Figure 2.4: An example of OS data restructuring
(node detection along a terrace front)



Unstructured DMB
vectors



Restructured PMA vectors
in link and node format

In summary, the restructuring project adopted in the PMA software has involved the following:

- a. Taking the OS internal data format as input, all records are sorted and reorganised so that all information pertaining to each 100 metre square on the National Grid is held together as a series of sub-directories for each square. In passing, studies by the author have shown that plotting time for the sorted (restructured) data are vastly improved; there is much to be said for sorting all vector-based geographic data before plotting.

Preliminary tests have shown current DMB data to be only 35 - 40 per cent efficient i.e. of the total pen travel distance (in pen up or down position), only 35 - 40 per cent of the travel is pen down.

- b. Working sequentially on each 100 m square sub-directory in turn, the intersections of all line features are detected and stored in a secondary file. These intersections (in PMA terminology) are the nodes of the data.

- c. All support-data such as textual information are separated from the line data and stored as separate data entities.

2.7.1 The file structure of the PMA restructuring process

The restructuring software now implemented by the Ordnance Survey generates a data structure suited to the ICL hardware installed in the headquarters. Random access memory (disks, drums, etc.) is somewhat limited for data storage and hence much use is made of sequential storage devices in the form of computer compatible tapes. All PMA data files have adopted a serial access thereby reducing costs of data storage but

increasing the storage and retrieval access times by a (presently) unknown factor.

The methods of data organisation incorporate three types of 'low-level' data file. These hold the 'atomic fragments' of the more complex DMB ground features as a series of points and lines (or nodes and links). 'High-level' files can then be constructed specific to user requirements by aggregating accordingly only those links necessary to represent the features requested.

The three low-level files are (PMA 1978):

- point and line data file;
- text index file;
- text data file.

These will now be discussed separately.

1. Point and line file: The point and line file for each map sheet stores the bulk of the topographic information; it is here where the links and nodes data are stored. The file is sub-divided into three sections: the grid nodes,

the internal nodes and the links.

- a. The grid nodes are a special category of point which are generated by the software each time a link crosses from one 100 metre grid square to the adjacent square. In this way, it is possible to logically chain features when they pass from one sheet to the next or from grid square to grid square within the same sheet.
- b. Internal nodes are those points described by Cartesian coordinates in which one or more links terminate. Links representing "wiggly" features can comprise many points but these are not nodes unless the link (or any other link) starts or ends there.
- c. The links are the lines which join together the nodes to represent series of topographic features.

The point and line file holds these three categories of data in the above order; since it is built up as a sequential structure, then very little topology can exist. Each node point has pointers to its associated link data although no positional information is included; the x,y points comprising the line-work are held in the

links section only.

2. Text data files: The textual information is held in two file types: The Text Index file, essentially, speeds up access to the Text Data file which holds the strings of text characters themselves. The index file comprises a series of records of varying length; each record provides a pointer to one or more text items in the text data file. The index entries are sequenced by grid square order. The text data file stores the text strings together with their position, style, height, etc.

2.7.2 Uses of the restructured low-level data

The restructuring strategy adopted by PMA has been to take on two phases of development. Phase one involves the actual restructuring process as described above. After considerable ^{problems,} this process now works sufficiently well to provide a limited customer service although it is a fairly time-consuming operation (OS pers comm 1979). Phase two of the strategy has involved the development of a User-language which can interrogate the low-level data and reconstruct user-defined entities such as land parcel boundaries, railway or water courses, administrative boundaries, and so on; these are

termed 'high-level' files. Thompson (1978) points out that initial trials at writing the user-language software have attempted to rebuild the link and node data into land-parcel boundaries; these have direct relevance to H.M. Land Registry and Local Authority property records and are considered to be the most arduous to develop. A number of technical and conceptual problems do exist in any entity rebuilding: most other systems generating polygons (e.g. GIMMS (Waugh 1979)) are aided considerably by additional information which OS data do not provide, namely what is to the right and left of each line. Furthermore, the decision to retain the characteristics of the original data as they have been digitised, although understandable, is undesirable in a data base context. Two lines joining at a common point, for example, need not necessarily have exactly identical coordinates for their end-points in the Ordnance Survey schema.

Ideally, the high-level files should be customised to suit user needs with zone of interest, data classification and any supplementary information (e.g. centroid reference, and road widths) being requested at run-time. Further, there has been much call for versatile software to run on user's installations as and when required. Sadly, some of these intentions have had to be deferred in practice: the user language cannot be considered as transportable, it is highly

dependant on the ICL 1906S running under the GEORGE 3 executive at OS headquarters. For users who wish to specify their requirements in detail on a once-and-for-all basis, there should be few problems; Ordnance Survey should be able to provide a magnetic tape of the restructured data comprising property and other zones specific to the user requests, although existing software is far from robust. This has been the case with the Dudley Metropolitan Borough which is attempting to use restructured land parcel data in their Local Authority Management Information System (LAMIS - see Harrison 1979); OS are digitising the required sheets in the production flowline and restructuring the results via the user language in house. As yet there remains substantial unsolved problems where users wish to restructure their own copies of the data, or indeed up-date or generally combine them with their own data sets. Arguably it may be the role of the national mapping agency or some other external institution to provide this service. If this is the case, then there is a need for substantial developments in standardisation of exchange data formats and documentation on the types of data manipulation possible; these developments are allied closely with the needs of users as outlined in chapter one.

2.8 THE USE OF FAST-DIGITISING SYSTEMS AT ORDNANCE SURVEY

The elapsed time to encode digitally all 220,000 large scale plans for national coverage is expected to be approximately 60-70 years with current throughput rates for the existing production flowline at Ordnance Survey (Logan 1981). In order to provide a National Digital Topographic Data Base in a much more reasonable timescale (the Serpell Report (HMSO 1979) suggests ten years) it becomes necessary to look towards faster, semi-automatic data capture methods. In 1979, The OS took delivery of a Fastrak line following digitiser manufactured by Laser-scan Laboratories Limited of Cambridge, England. This highly advanced digitising hardware has formed the front end of the TODDS (Topographic Data base Development System) project (Logan 1981) and it is hoped that such an approach to the data capture stage will increase throughput rates dramatically (c. five-fold). The hardware, which is highly ingenious, involves the use of a focussed laser beam which detects light and dark differences as it probes areas of film to be digitised. The transition from light to dark or dark to light represents the edges of the line to be digitised; appropriate driving software instructs the internal deflection system how to move the laser to follow logically each line automatically (Hardy 1978). The device operates in a fully interactive manner, stopping for operator assistance if it cannot see any logical continuation

of the line being followed; data capture halts at the end of open features or when it returns to the start of a closed loop feature. Operator intervention enables feature coding and classification to take place at the digitising stage on a similar basis to the manual tables; a keyboard console and tracker ball is used to indicate the class of feature (point, curved line, pecked line) and its position. The operator works in front of a large 1.0 x 0.7 metre display screen (based on the Laser scan HRD1 laser display/plotter) on which is projected the map data to be digitised. Any limitations stem largely from the photographic reduction process of the source document and hence loss of quality and reduced line weight; the readable area of the fiche is 98 x 68 mm. Fastrak has the added benefit of being able to detect the junctions of two or more lines ('nodes' in the PMA schema) and can, therefore, be programmed to digitise directly into link and node format, thereby removing the slow and costly restructuring stage (for OS, if not for customers who wish to link their own, existing digitised data to that in the OS files).

The first production Fastrak to be installed in the UK was at the Mapping and Charting Establishment (MCE) of the Royal Engineers in March 1977 and reports from there have been very favourable (Howman and Woodsford 1978) for small scale topographic work. The device is said to retain

cartographers' interest and the error rate is low. They report the clean digitising of 21.6 metres of contour data in one hour. Digitising rates for more complex large scale and land-use maps are inevitably slower. Hardy (1978) has suggested a useful empirically derived formula for approximate digitising times on a variety of map types. This is:

$$T = \frac{L}{12.5} + \frac{N}{720}$$

Where: T = approximate time in hours
L = line length in metres
N = number of features

Time estimates are subject to variations due to operator skill and assumes the use of simple numeric feature codes only. Appendix 2.1 shows similar expressions for manual digitising procedures for comparison purposes. Assuming a typical OS 1:1250 map sheet containing 40 metres of line length and 2100 features (see Adams 1979 p 44-45) then estimated digitising time on the Fastrak would be six hours. Manual digitising of the same sheet (from Ordnance Survey records) show a time of 24 hours; thus, Fastrak represents a four times increase. Clearly, digitising rates will be sheet-specific but four to five times increases are to be

expected and many preliminary tests (Howman and Woodsford 1978) suggest this to be probable.

In addition, the TODDS project has involved the trial of two interactive graphics work stations (Logan 1981). One station supports the IGES software (Laser-scan Laboratories 1981) for map data verification and manipulation. The other station supports the CLUMIS system (Ferranti-Cetec Ltd. 1979) to edit land parcel boundaries after the digitised data have been restructured into links and nodes. Since the hardware involved with both these systems is identical, it is easy to select any combination of systems running simultaneously.

It has been anticipated that trials with the TODDS system over a period of 12 - 14 months should give some indication of how such a system will perform in a production environment. Initial trials have involved examining alternative flowline procedures and comparing costs and throughput rates (Logan 1981). Results at the end of 1981 have suggested that, by incorporating Fastrak digitising and interactive edit facilities into the digital flowline, there can be favourable developments: costs will be reduced substantially for the initial generation and maintenance of the data base.


2.9 DATA BASE MANAGEMENT SYSTEMS AND TOPOGRAPHIC DATA

The 'relational data base' concept as suggested by E. F. Codd (1970, Chamberlin 1976) have been shown to be widely applicable to commercial data base management systems. These methods of data organisation aim to optimise data storage and information retrieval times. The fundamental concepts involve a data structure comprising a number of files or 'relations' which are equivalent to a table of figures -- see figure 2.5

Figure 2.5: The structure of a relational data base

a TUPLE is equivalent to a row in the table

a DOMAIN is equivalent to a column or a series of fixed length records

 = Primary key

In order to enable cross-referencing of tuples in different relations within the data base, it is necessary to assign to each tuple a 'primary key' to define and identify

it uniquely. In this way it becomes possible to implement the relational algebra theory developed by Codd for efficient data manipulation and retrieval. Since relational theory assumes random access file storage methods, then it is necessary to incorporate magnetic disk or similar storage media into the system.

Baxter (1980) has attempted to modify his DOE (1976) work involving a parsed data set of British Local Authority Boundaries to utilise a relational type of data structure; he anticipates much improved retrieval overheads and times for user-selected polygons.

In relational terms, the current file organisation of the restructured OS data is similar to the schema outlined in figure 2.6.

The description in figure 2.6 is an attempt to define the PMA restructured data files into a relational data base specification. The result is far from acceptable; variable length tuples evident in every relation are almost certainly invalid in the Codd schema. The need for tuples of variable length arises from the non-constant number of links associated with each node. The best form of data organisation suited to these characteristics is probably a tree structure as shown in figure 2.7, although each sub-file

(i) Grid node relation (G):

S#	Grid edge	No. of associated links	
----	-----------	-------------------------	--

Information about each assoc. link (i.e. for every link give: grid sq., data division, serial no., feature code, azimuth)

In Codd (1970) relational terms: $G(S\#, GE, NAL, \dots \text{variable length} \dots)$ where: $S\#$ = serial number (prime key)
 GE = grid edge
 NAL = no. of assoc. links.

(ii) Internal node relation (I):

S#	Grid square	No. of associated links	
----	-------------	-------------------------	--

Information about each assoc. link (i.e. for every link give: data division, serial number, azimuth, feature code)

In relational terms: $I(S\#, GS, NAL, \dots \text{variable length} \dots)$ where: GS = grid square

(iii) Link relation (L):

LS#	Grid square	Feature code	Data divn.	Azimuth of each end	SW node no.	NE node no.	No. of coords
-----	-------------	--------------	------------	---------------------	-------------	-------------	---------------

x,y coords. that generate the link

In relational terms: $L(LS\#, GS, FC, DD, A(1), A(2), S(1), S(2), NOC, \dots \text{variable length} \dots)$

where: $LS\#$ = link serial number; FC = DMA feature code;
 DD = data division; A = azimuth of ends of link;
 NOC = no. of coords comprising link.

(iv) Text relation (T):

T#	Grid square	Data divn.	Data subdivn.	Start pt x,y	Height	Style	Spacing	Azimuth	No. of chars.
----	-------------	------------	---------------	--------------	--------	-------	---------	---------	---------------

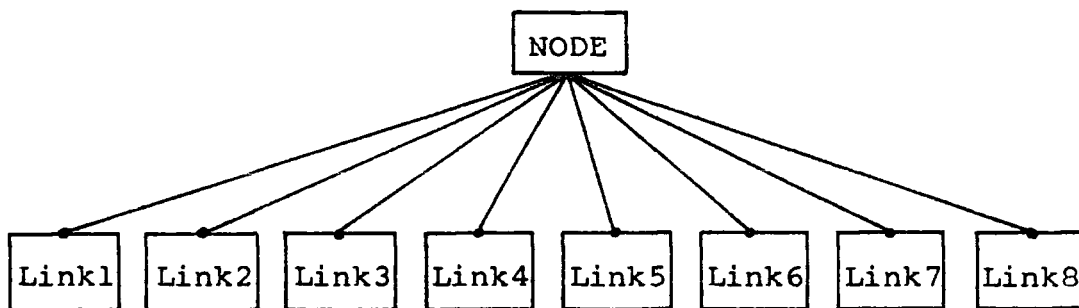
text string (variable length)

In relational terms: $T(TS\#, GS, DD, DSD, TXY, HT, ST, TAZ, SP, NTC, \dots \text{text string variable length} \dots)$

where: $TS\#$ = text serial no; DSD = data subdivision; TXY = text start posn.; HT = height; ST = text style;
 TAZ = text azimuth; SP = text spacing; NTC = no. of text characters.

can be treated as 'chains', 'stacks', or 'rings' (these are terms used in computer data structure theory - see Page and Wilson 1978).

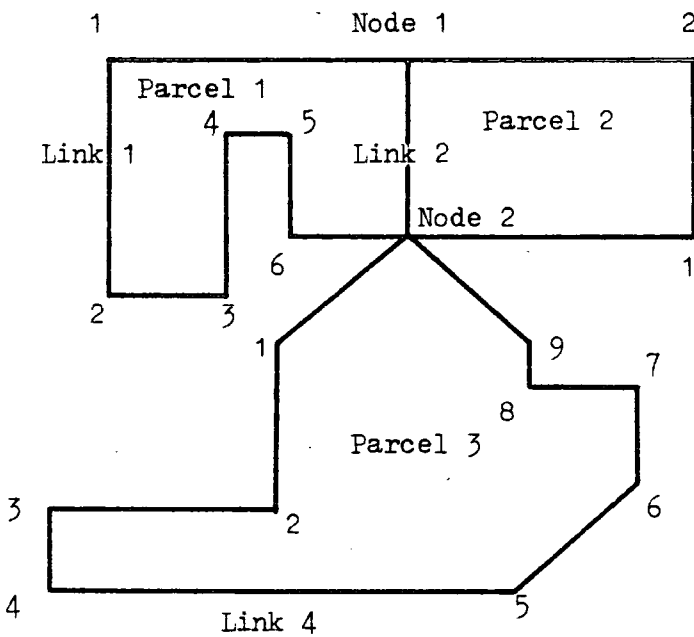
Figure 2.7: A tree structure to store link and node data



The coordinate strings within the link sub-files will always be processed in the same order; some links close back on themselves suggesting a ring structure, whilst others have two separate end nodes suggesting a stack. An added complication is introduced since a node may be accessed by more than one link, suggesting a tree structure to be inappropriate; a more complex multi-directional network (or plex-network after Martin 1976 page 84) is required. Further, the 'rings' or 'stacks' will need to be accessed in both directions (ascending and descending order) suggesting the necessity for multi-directional data files.

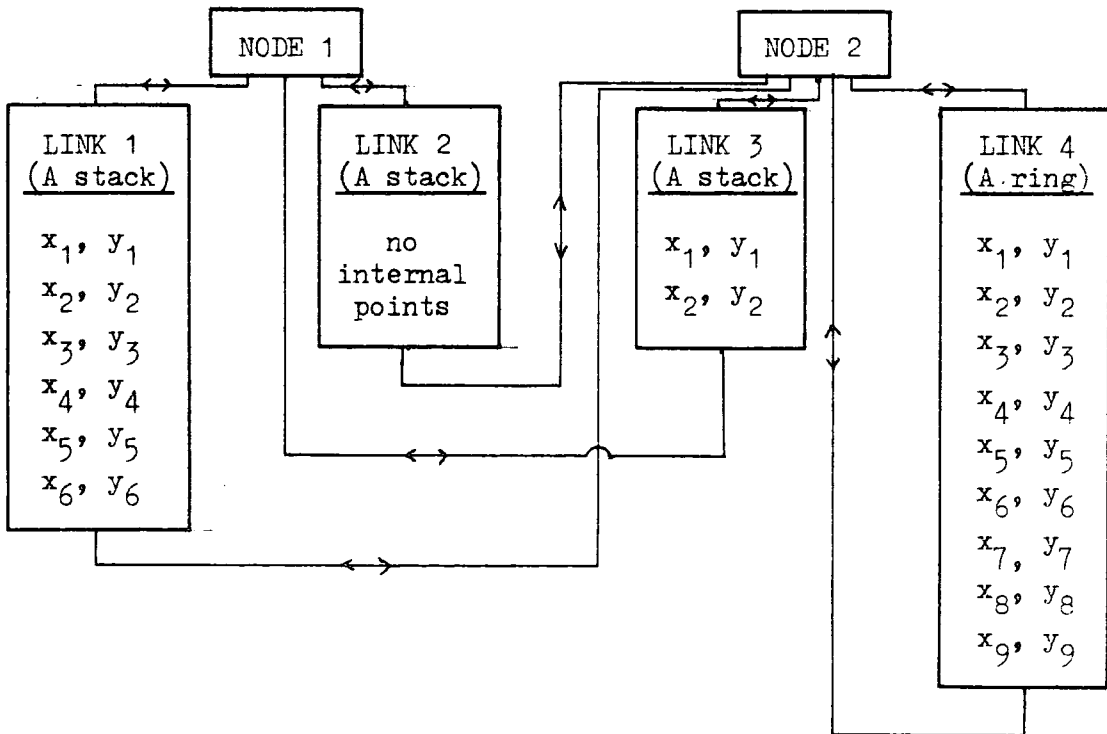
Consider an example shown in figure 2.8:

Figure 2·8: An example of the complex data base schema
required to represent the simplest of
vector link and node structures



(a) Reality:
 3 land parcels,
 involving:
 2 nodes,
 4 links.

(b) The data base schema for the above:



Figures 2.8a and 2.8b portray how even the simplest of situations require a complex schema. Relational ideas can offer a better solution if the ring or stack sub-files are incorporated into the system with some form of hash-addressing (see Page and Wilson 1978 page 174). Since it is the link sub-files which give rise to tuples of variable length, then these can be held as hash addressed stacks; the node serial number in conjunction with some clockwise sequence link number at the node can provide the seed for the hashing algorithm.

Figure 2.9 shows the incorporation of hash addressed stacks for the links in a suggested relational data base for restructured OS vector data. An alternative approach for polygonal data has been suggested (Harris et al 1981) which holds the data in the form of hierarchical levels: at the lowest level is the individual line segments with appropriate link and node data stored higher up the hierarchy.

2.10 THE UNITED STATES GEOLOGICAL SURVEY ALTERNATIVE FOR VECTOR-BASED DIGITAL MAPPING

At the outset, it should be pointed out that mapping practices and user needs in the United States differ quite dramatically from those in the United Kingdom. However, USGS

Fixed-length
Relations

Variable-length
Stacks or Rings

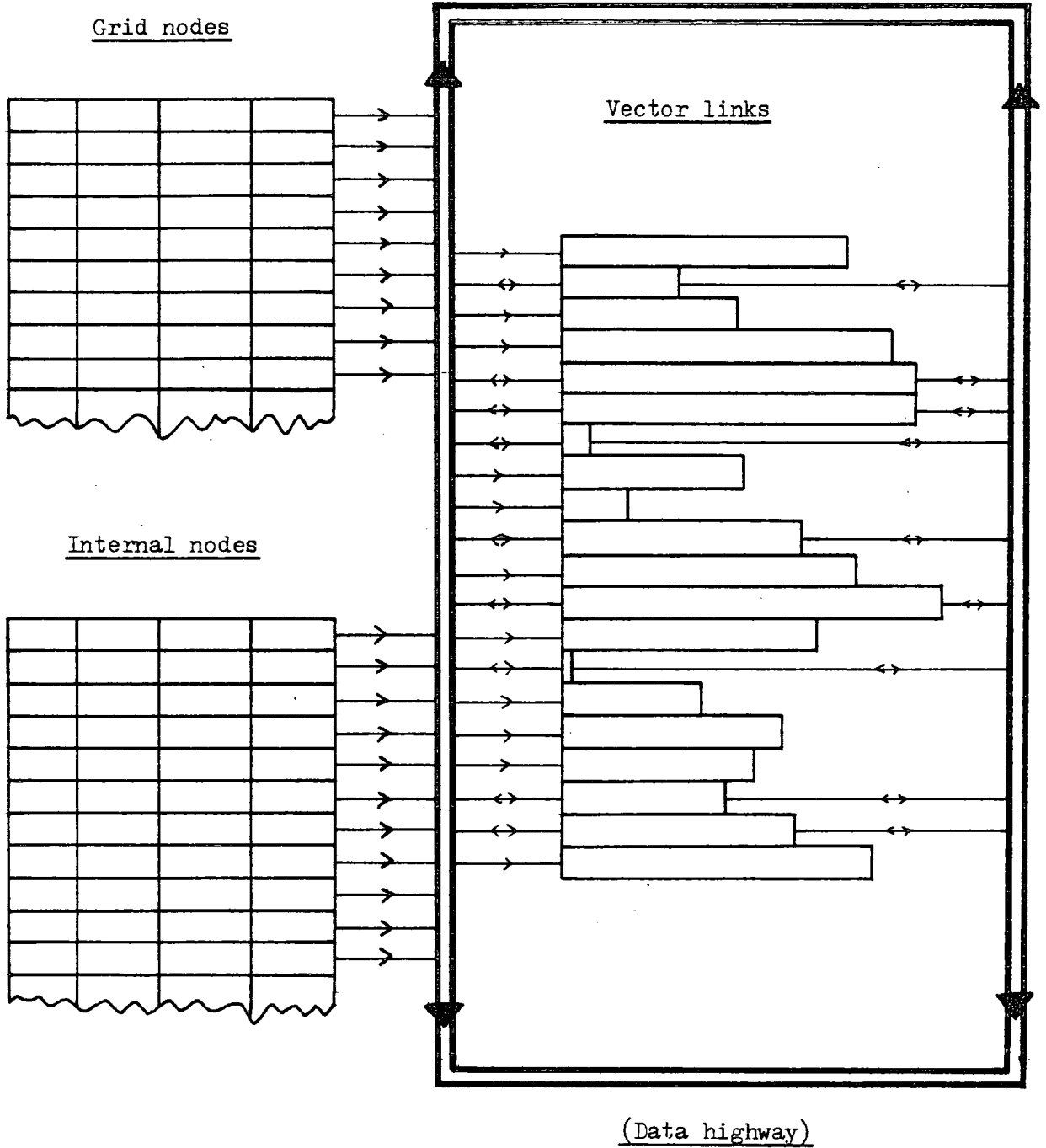


Figure 2.9: The Ordnance Survey restructured vector data base

policy on digital mapping is worthy of note since it suggests an interesting and potentially more appropriate alternative to vector data structures and user needs.

2.10.1 The role of the National Mapping Division of USGS and the National Mapping Program

There are now four mapping centres in the USA and these are based at:

Merla Park, California (Western)	325 Staff
Denver, Colorado (Rocky Mountains)	325 Staff
Rolla, Missouri (Mid-Continent)	325 Staff
Reston, Virginia (Eastern)	360 Staff

The objective of the National Mapping Program is to prepare and make available multi-purpose maps and fundamental cartographic data to meet the requirements of users throughout the United States (McEwen 1980). The basic mapping scale providing complete coverage of the country is 1:250,000 - this comprises 468 map sheets (625 including Alaska and Hawaii). The largest scale with which the

National Mapping Division (NMD) is involved 1:24,000. Approximately 600-800 new sheets are produced each ^{year} and are revised on a cycle between five and fifteen years (dependant on the priority of the area; Washington DC, for example, is revised every five years). There are 54,000 sheets in the 1:24,000 scale series; seventy per cent of these had been published by late 1980, 21 per cent of which had been produced as fifteen-minute quads. Other scales currently being produced are the 1:250,000 scale land-use series showing information at ten and forty acre resolutions (see Rhind and Hudson 1980), the 1:100,000 series, of which there are 1800 sheets, and a 1:2 million scale National Atlas.

A major activity of the National Mapping Programme is the development of cartographic data in digital form with special emphasis towards digital coverage of the 1:24,000 series. A secondary activity is to provide a small scale data base for applications such as special graphics products, gazetteers and index generation.

The recent hardware acquisitions for the digital products are itemised in McEwen (1980). Much more reliance is being placed on photogrammetric methods than in the Ordnance Survey system, which is reflected in the famous orthophoto maps emanating from the USA. Nine analytical stereo plotting systems have been acquired based on Kern PG2 and Wild B8

stereo plotters and Gestalt GPM-2 photomappers. The need for interactive data edit facilities has been seen as crucial to handle most needs; each station (produced by M & S Computing Inc.) has a digitising table, edit tablet, dual Tektronix 4014 graphic display screens and an alphanumeric console.

2.10.2 The data-structure to be adopted

The National Mapping Program makes a clear distinction between the uses of Digital Elevation Models (DEM) and Digital Line Graphs (DLG), both of which are seen as important for satisfying user needs; Ordnance Survey only concerns itself with the latter. McEwen (1980) points out how a hierarchical conception of the DLG's can satisfy most applications:

- DLG1 - line map information collected and coded to prescribed standards and edited to remove acquisition blunders;
- DLG2 - DLG1 + additional attribute codes and removal of visual errors and inconsistencies;
- DLG3 - DLG2 information that has been spatially

structured to define all topographic relations.

DLG2 has only sufficient 'intelligence' for the production of graphic output and is comparable with the Ordnance Survey DMB production system. For any geographical information system (GIS) support, DLG3 data are required and current pilot projects have been established to collect 1:2 million data in DLG3 format. The generation of a data format with such high degrees of structure requires a highly ingenious data base system and a mammoth digitising task. Most of the topology of the 1:2 million mapping is being built into the data structure at the capture stage in the form of multi-attribute codes (Stephens 1980); all digitising is being carried out by hand on conventional moving cursor tables. As lines are followed, flags are added to show on which scale of mapping they should appear; the thresholds are dependant on such factors as line length, and polygon acreage as well as taking into account the smoothing and filtering of points. Additional attributes are held for each feature, such as rail tonnage information, road types, forest lands as well as size, and pointers to adjacent features.

The information is held in the data base as vectors and is aggregated on a sheet by sheet basis. By mid-1980, twenty one sheets of the 1:2 million scale mapping had been

completed: there are seven overlays per sheet to include all the data types.

This project is the first of its kind. On no other digitising flowline (c.f. ECU geological coding on experimental mapping production in the early 1970's) has so much additional information been built into the data from the outset. Experience has now shown that editing requirements were initially high, chiefly as a result of inexperienced personnel. These inefficiencies have gradually been reduced as more experience has been gained by the digitising staff and as a result of careful tuning of the system.

It has been argued earlier (section 2.4.2) that a knowledge of the likely characteristics of projected topographic data bases in digital form is of practical importance. USGS readily agree that the projected characteristics on data quality and volume are essential for an orderly and productive growth (USGS pers comm 1980). Generation rates for their digital cartographic data will reach alarming proportions, typically 70,000 megabits of information per year or more from the DEM data alone.

2.11 CONCLUSIONS

The USGS digital mapping projects are a first indication of attempts to provide a flexible computer mapping system that is not dependant on scale. In most cartographic data bases until recently, the arrangement and uses of data are guided by the input stage (Peucker and Chrisman 1975); little manipulation is attempted once the data have been captured. The most economic form of data storage will depend entirely upon the characteristics of the data themselves and the representation used for storage. Vector data structures have suited these representations well because of the inherent nature of information presented on maps; the conception of connected points to represent a line is straight forward and well tested. For geometrical shapes (e.g. curves of railways), the use of widely separated digitised points connected by splines or similar functions is manifestly possible. Indeed many ingenious data structures have been considered and tried in an attempt to provide more intelligence to the resulting system. More paradoxically, very little attempt has been made to combine different types of cartographic information, for example census data or satellite imagery, with topographic data. This is almost certainly a result of inadequate and incompatible data formats with costly conversion processes being necessary to alleviate the problem.

APPENDIX 2.1

Derivation of a digitising time prediction formula for the
Ordnance Survey DMB digital flowline by least square methods

Assumption: The time taken to digitise an OS large scale plan by manual methods is dependant on the total line length and total number of features.

$$T = aL + bN$$

where T = digitising time in hours
L = total line length at
scale in metres
N = number of features on map
a,b = constants

Theory:

$$T_c = \frac{aL}{c} + \frac{bN}{c} \text{ -----(1)}$$

Let the suffix c represent the correct or true value
Let the suffix e represent an experimental value

There will be a residual value between the true value of T and the experimental value of T, let this be r.

$$r = T_e - T_c$$

from (1): $r = T_e - (aL_c + bN_c)$

i.e. $r = T_e - aL_c - bN_c$ -----(2)

The least squares solution stipulates that the sum of the squares of the residuals (s) must be a minimum.

$$s = \sum_{i=1}^{i=n} (T_{ei} - aL_{ci} - bN_{ci})^2 \text{ should be a min ---(3)}$$

The conditions for s to be a minimum are:

$$\frac{\partial s}{\partial a} = 0 \text{ ----(4) ;}$$

$$\frac{\partial s}{\partial b} = 0 \text{ ----(5)}$$

From (3) and (4):

$$\begin{aligned} \frac{\partial s}{\partial a} &= 0 \\ &= \sum_{i=1}^{i=n} 2 (T_{ei} - aL_{ci} - bN_{ci}) (-L_{ci}) = 0 \end{aligned}$$

Divide both sides by -2:

$$= \sum (T_{ei} - aL_{ci} - bN_{ci}) L_{ci}$$

Expanding:

$$= a \sum L_{ci}^2 + b \sum L_{ci} N_{ci} = \sum L_{ci} T_{ei} \text{ ----(6)}$$

Similarly (3) and (5) gives:

$$a \sum_{ci} L_{ci} N_{ci} + b \sum_{ci} N_{ci}^2 = \sum_{ci} N_{ci} T_{ei} \text{ --- (7)}$$

Equations (6) and (7) are solved for the two unknowns a and b.

The values of L and N were available in computer form from a study of data characteristics (Adams 1979). The values of T were extracted from Ordnance Survey records at the Southampton headquarters.

By computer program:

$$\text{For 1:1250 data --- } T = \frac{L}{1.75} + \frac{N}{144.1}$$

$$\text{For 1:2500 data --- } T = \frac{L}{1.10} + \frac{N}{128.2}$$

The effect of the underlying assumptions in the theory

The Statistical Package for Social Scientists (SPSS) was used to investigate the Pearson correlation coefficients between N, L and T.

1:1250 scale		N	L	T
	N	1.0000	0.9389	0.6786
	L	0.9389	1.0000	0.6453
	T	0.6786	0.6453	1.0000

1:2500 scale		N	L	T
	N	1.0000	0.8844	0.7679
	L	0.8844	1.0000	0.6930
	T	0.7679	0.6930	1.0000

These figures, together with scattergrams suggest that digitising times cannot be wholly derived from line length and feature frequency. The rates of output from different operators are bound to vary and this factor was not taken into account. However, the results are of, at least, some use; they show that some use of planning is certainly better than no method at all.

3 RASTER DATA STRUCTURES

3.1 INTRODUCTION

As we have said in chapter two, the digital representation and storage of spatial data in a computer has traditionally tended, in Britain, to adopt a vector-based method of organisation. Chapter two described the current state-of-the-art in vector technology with the aid of various examples from working and developing systems. Recently, increasing attention has been focussed on alternatives to this now well-established approach; it has been argued (Peuquet 1979a) that raster data structures have much potential in mapping and spatial analysis. This chapter defines the distinctions between vector and raster technology and considers some of the applications to which data in these raster forms can be put.

The term 'raster-format' has come to be associated with a grid-based representation of spatial information which has its origins in video transmissions (television), raster scanners used in remote sensing and in similar disciplines. Conceptually, a graphic image can be represented as a matrix of picture elements or pixels, the size of which determines

the resolution of the image. A raster-based representation of spatial quantities can be achieved by a number of means: the simplest of approaches will involve some form of manual grid sampling, although more appropriate encoding methods have been connected with analytical stereo plotters (Burnside, 1979 page 231), satellite imagery (Barrett and Curtis 1976) and optomechanical scanners (see chapter seven).

Vector representations of graphical data, then, are most appropriate for the storage of linear features: the technique provides for efficient use of backing store and conventional graphics devices for output. Such an approach, however, would be highly inappropriate for the representation of most photographic images or hand-drawn paintings. By their very nature, series of contiguous picture elements are more appropriate for the portrayal of these types of data - i.e. an 'area-fill' rather than 'boundary-drawing' approach. Indeed, the adoption of a grid based notation for the storage of many geographic variables within a computer system can expedite easier manipulation and flexibility according to Peuquet (1979a): counts of population, frequencies of old age pensioners, mortality rates, unemployment data, etc., can be held on a grid square basis (though they are only rarely made available on this basis - see Rhind 1981). Ratio information can be determined logically and easily in software by performing arithmetic operations with two or more arrays of

information on a grid square-by-grid square basis. Clearly great care should be exercised on the type of measurement scales being adopted. Ratio data could no longer be valid for arrays of land use codes, for example, since the data are no longer on an interval scale but are essentially nominal in nature. It will be possible, of course, to manipulate the data in other ways and obtain frequency counts, dispersion statistics and the like; for some applications of this nature, notably those involving spatial operators, quite complex data representations are required when the data volume is anything other than trivial.

3.2 RASTER DATA REPRESENTATIONS

The notion of resolving images into a series of (usually square) digital picture elements or pixels was described in chapter one. In fact, grid formats were frequently used for output purposes in the early days of automated spatial data manipulation and presentation, long before vector plotters became generally available: computer mapping by line printer was commonplace in the early sixties. SYMAP (Schmidt and Zafft 1975) is the most widely used mapping package of this variety where each grid value is easily translated into its associated line printer character. In the simplest topographic case, each pixel can take on one of two binary

states (on or off) - the so-called bit map since each data element can be represented by one binary digit only. Under these conditions, the storage overheads will be a minimum although the characteristics of the graphic being represented will be necessarily limited since each data item will be restricted to one of two distinct states only. This is inappropriate for the amounts of data inherent in the Ordnance Survey large scales mapping. In such circumstances there can be no knowledge of what the set (or black) cells represent and hence this type of notation is necessarily limited in its applications. Bell Laboratories in the United States are finding bit map data highly applicable for the first stages of digital representation of technical and engineering drawings (Pferd, pers comm 1980) in part because their immediate need is to transmit details of such graphics documents over telephone lines. Similar strategies were the means used to encode more than one thousand maps by the Canada Geographic Information System: the 'meaning' of the cells was provided at a later stage (Tomlinson et al 1976). Other applications for bit map storage could involve instances where the location of only one variable type on a nominal scale is being displayed; a common situation in data being represented by dot maps and some simple choropleth mapping which shows only two item types. Colour separation masks are essentially binary maps and could be held as bit maps, as could representations of areas which meet binary

conditions such as 'areas within 1 km of a main road and over 300 metres above sea level'.

Within fairly stringent limitations, the bit map provides an appropriate strategy for storing certain data sets with manifestly simple characteristics. In order to represent digitally the more complex types of information, such as the examples of population data cited earlier, it then becomes necessary to add further information (or increased intelligence) into the coding schema. In essence, this requires the capability of attaching additional detail to each data element over and above the on/off state. This can only be achieved by allocating a greater number of binary digits for each raster element; the number of bits reserved will increase storage requirements and slow retrieval access times. Without some specification of task and of efficiency requirements at the outset, data volumes could reach alarming and unmanageable proportions. Multi-sectral data emanating from Landsat 1 and 2 imagery, for instance, is held in raster form providing up to 128 different values of grey levels: one byte of eight bits is allocated to each cell in the export data format (Harris 1979). With data from these satellites being updated every eighteen days then the volumes being generated are clearly substantial: some 7.6×10^6 bits of information may be needed to store one Landsat frame, and four such frames (in each of four wavebands 4,5,6 and 7 - see

Barrett and Curtis 1976) are produced for every 185 x 185 kilometre scene. The complexity of the coding scheme (in this case the number of grey levels which it is possible to distinguish) is undoubtedly an important governing factor in the volume characteristics of a data set. Yet, the fundamental factor which governs storage requirements is the resolution of the pixels. Clearly to reduce the number of grey levels possible in a Landsat image by a factor of two would not halve the storage requirements for an image since this could be achieved by merely removing the most significant bit from each pixel. A specification of the resolution of each pixel is equivalent to describing the number of pixels necessary to represent an image. The increase in storage overheads consequent on increasing resolution will rise as the square of the resolution. As one

the Thematic Mapper on Landsat 4 processes data with a resolution of 30 metres, although images are coded as 8 bits per pixel, there are seven channels in total. Thus, Landsat 4 TM images have 10 times as much data as equivalent MS data, have 6 times as much data as

A second example concerns GEC-Marconi Ltd., a large electronics group in the U.K., who have developed a highly specialised raster scanner to simulate a data source for side-looking airborne radar in laboratory conditions. The pixel resolution is fixed at around 13 microns which in 1981 was generating severe storage problems: a 1.5 centimetre square area filled a dual density floppy disk of 500,000 bytes.

The incorporation of colour information, or indeed any feature coding techniques, into the list of attributes is a logical extension of the grey level technique. A multi-coloured graphic with up to sixteen different hues can be represented in a raster-format providing that each cell is assigned four bits. In principle, this is inappropriate for data with more than sixteen classification types although could still have potential if no more than sixteen different feature codes need to be examined at any one time (cf. Scitex Response 250 - see chapter seven). To represent digitally the 109 feature codes in the schema of the Ordnance Survey large scales DMB flowline in a raster form will require at least one byte per pixel; this would actually provide up to 128 different feature categories assuming one bit is reserved for system use such as 'parity validation'^{or negation}. This provides a good example of the need for careful system design to meet present and potential user needs. Without some clearly defined specification at this early stage, quite catastrophic problems could develop. It would, for example, be a major reformatting task to extend to more than 128 feature types once a single byte has been allocated to each

multi-spectral scanner in an environmental remote sensing satellite since these are usually restricted to some set

from the adder array; 10-bit data are displayed finally on the hardware's screen.

A similar case is likely to evolve with 'artificially coded' pixels representing user-defined features within a data base structure. In such circumstances, attempts are made to build more and more intelligence into the data set as experience and user needs become more clearly defined. This has been the case with the existing vector digital encoding methods used by the Ordnance Survey; the initial catalogue of eighty feature types in the prototype DMA system has necessarily risen to over one hundred in the improved (DMB) replacement system to cater for user needs. The ability to increase the list of feature codes in a raster topographic data structure is less simple. Once a set number of bits has been assigned to each pixel, then to increase this number will involve more complex processing and possibly require specialised hardware. Fortunately, the list of feature codes developed for the existing vector files are equally relevant to the raster files and hence the variety of feature types is unlikely to alter dramatically after some ten years of evolution unless the specification of data collected by Ordnance Survey is changed. Nonetheless, the importance of pre-defining data specification and tasks from an early stage cannot be over-emphasised. Such factors, ^{though,} do not have so much importance in the _{raw} digital representations of remotely-sensed data by satellite or by terrestrial photogrammetry; in a monochrome photograph for example, there is a finite number of discernably different grey levels. Similarly, half-tone technology in the printing industry need

only have the ability to handle limited contrast variations (Keates 1973, p 101) to be sufficient for most conditions. Mueller (1970) and Rosenfeld and Kak (1976) discuss the problems of the human perception of contrasting grey tones in graphics: it is generally believed that as few as eight grey levels is all that the human optical system can distinguish ^{at any one time.}

In remote sensing, though, this is only the case for the raw, input data generated by the sensors. In passing, new sensors are likely to provide a variety of radiometric resolutions - typically 6, 8 and 10-bit data. The subsequent image processing carried out on these raw data then generates more useful, more complex data with varying bit-lengths per pixel. For example, the subsequent processing is likely to rely not on visual discrimination, but rather on multi-dimensional statistical discrimination. This is likely to be just as unpredictable as for field generated topographic data. However, the basic data archive for remote-sensed data should remain the raw input data with fixed-length pixels.

It is suggested that a similar case is likely for raster topographic data. However, far less complex image analysis of such data is likely to be necessary if the total list of feature-types required to describe the 'conceptual image of the environment' is established at the outset. This is difficult to predict, though, unless a stable classification has evolved and to some extent, complete coverage of the country has been achieved. Without this, the list of attributes will tend not to be fixed and must evolve to meet every situation. This has serious implications for automated data entry into a digital topographic mapping system; more codes are required to describe the conceptual data base than is generated in data emanating from a scanner which sees only a physical image.

3.3 THE DATA CAPTURE STAGE

The conversion of a graphic format to a grid or raster-based digital representation for large data volumes must involve the use of some form of photo-mechanical scanning device. Any hardware design will be centred around a modulated light source which is scanned over the graphic in some way to detect the intensity and wavelength of light being reflected or transmitted. The choice of light source and the principle of operation is becoming more and more sophisticated as ingenuity and experience advances. Light sources used to date include: panchromatic white light, cathode ray beams, laser beams, quartz halogen or plasma sources or light-emitting and laser diodes; each has advantages and limitations on such factors as ease of control and modulation, maintenance, lifetime and durability and band-width precision. Such important technical characteristics as these, added with the overall system design which controls the scanning operation, must be carefully considered to achieve optimum performance. Important specification statistics for describing a scanner's performance are repeatability and accuracy; information which has hitherto been associated more closely with output devices. Repeatability is a geometric tolerance value within which the device will repeat its performance for given input data; figures of ten microns (10×10^{-6} metres) and better

are currently possible on modern scanners. The transport mechanisms for the source documents to be scanned have tended to take on either a flat-bed or drum-based design in a fashion similar to hardcopy vector plotters. Both are proving to be successful designs; examples are given in chapter seven of commercial systems using both methods with specifications which are acceptable for cartographic applications.

3.3.1 The drum-based principle of operation

Drum-based scanners operate on principles similar to the vector pen plotting hardware emanating from manufacturers such as Calcomp, CIL and Benson. The graphic to be scanned is attached to a rotating drum, sometimes by clips but more often on advanced models by vacuum pressure. As the drum rotates at a constant (synchronous) speed, a scan line in the y-direction is created. Incremental movement of this scan line in an associated x-direction is facilitated by controlled movement of the scanning head axially along the length of the drum, usually by some precision servo motor. Under the control of appropriate software, the drum rotation speed, the step-wise movement of the scanning head and the modulation of the light source can provide the required raster scanning characteristics to detect and provide pixel

data which describe the graphic under scrutiny; pre-defined thresholds and tolerances will be constrained by the hardware specification. ^{On photo-mechanical hardware,} these thresholds are dependant on the characteristics of the hardware such as the rotational speed of the drum, the incremental step movement of the axial head and the frequency modulation of the light source ^{or its returned signal}. Boyle (1979) points out that a rotational speed of 300 rpm (5 revs per sec) is commonly chosen by designers, which suggests that one scan line is output in 0.2 seconds. A total step-size of 10,000 increments of the axial scanning head over a distance of one metre, which is certainly within the bounds of technical capability, would thus suggest a total scan time of approximately 40 minutes.

3.3.2 The flat-bed principle of operation

Generally it has come to be expected that hardcopy vector graph-plotters designed around the flat-bed principle can produce computer plots with much improved quality over drum-based systems. This is not necessarily the case with raster scanning devices; systems do exist with very similar rates of performance using both design strategies. However, constructional costs can be lower in flat-bed scanners.

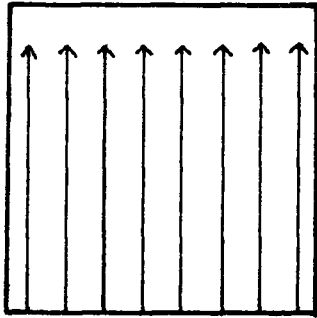
As the name suggests, a flat-bed scanner is based upon a table-top-like surface above which is the optical scanning head capable of moving in two mutually perpendicular directions. The dimensions of these devices tend to be larger than any drum-based equivalent and hence they are highly dependent on the availability of floor space. However, some special categories of scanners which take as their input-medium photographically reduced images on film or microfiche (eg the PEPR from the Image Analysis Group of Oxford University - see chapter seven) also adopt a variant on the flat-bed method of operation.

Generally, the graphics to be scanned are securely fixed on the flat-bed scanning surface by means of some form of glass pressure plate which can be raised or lowered. In principle, the scanner's optical head is then made to move over the surface of the graphic in a series of parallel swathes as depicted in figure 3.1a.

There are several advanced variants on this basic theme. For example, every alternate scan line can be scanned in reverse order as suggested in figure 3.1b; such an approach will increase throughput rates virtually by a factor of two. More recently, push-broom scanners have become serious design contenders. These essentially have the ability to scan several contiguous lines simultaneously; this variation can

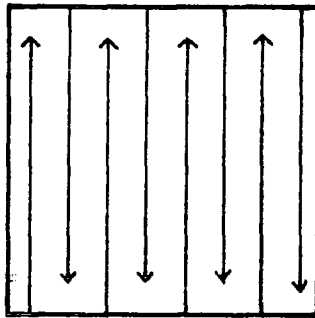
readily improve throughput rates by a factor of ten (Boyle 1979).

Figure 3.1 Some scanning strategies used by flat-bed scanners



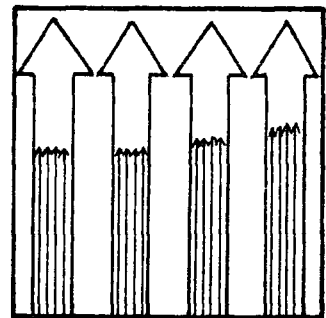
[a]

Omni-directional scan



[b]

Alternate direction scan



[c]

Push-broom scan

3.3.3 Appropriate pixel resolution

The choice of a suitable pixel resolution to represent an image is a question of much theoretical debate. A careful balance has to be established between a tolerable volume of data for subsequent computer processing and a sufficiently accurate depiction of a graphic in digital form as is necessary for the task in question. The resolution of a raster image refers to the number of picture elements per unit length of line, often quoted in inverse form. Values

can range to as fine as 13 microns for highly specialised work (eg the work of GEC-Marconi - see chapter seven). About 250 pixels per inch is not unreasonable for most graphic applications (Pferd and Ramachandran 1978, Tomlinson 1972) which suggests the spacing of the scan lines to be of the order of 0.1 mm (0.004 inch). Certainly the selection of this order of resolution is appropriate for accurately depicting documents that have been drawn by hand originally. Applications adopting twice as high a resolution necessitate a quadrupled rate of data handling and storage: hence, a resolution of 500 pixels per inch requires a four times increase and 1000 pixels per inch requires a sixteen times increase.

Generalisation of an image being represented by a raster data set is a function of the pixel size and care should be taken not to over-generalise the image for a given task. However, it is doubtful whether any benefits will be gained from a raster image with a pixel resolution that is higher than is necessary. The production of graphics and detailed manipulation, storage and transportation of the data will be far more arduous with finer pixel resolutions, although post-digitising generalisation is always possible.

Chapter three



3.4 THE IMPLEMENTATION OF A RASTER-BASED DATA STRUCTURE

The problems associated with the large volumes of data inherent in raster methodologies have been mentioned in passing. The concept of a bit map cited in section 3.2 provides less problems of storage and processing than does the ability to incorporate grey levels, colour or feature codes within the data. To represent a typical 40cm x 40cm Ordnance Survey 1:1250 plan with a pixel resolution of 20 lines per millimetre (0.05mm) requires an array of 8000 x 8000 elements. This suggests a storage requirement of eight megabytes to portray the information in binary form. To represent each pixel with a series of grey levels, as is the case with Landsat data, would necessitate the allocation of eight bits per pixel. This would expand the size of the complete raster buffer to 64 megabytes which is, essentially, larger than the early sixty megabyte capacity ICL EDS60 disks which were the standard random access storage device on ICL machines until circa five years ago. Generally, however, it is possible to treat the raster-based images as sparse arrays and compress them by several orders of magnitude to more manageable dimensions. An 'average' 1:1250 OS plan with the order of 2000 features and 40 metres of mapped line length has been found to be approximately 98 per cent sparse (i.e. 98 per cent of the pixels in the raster image are blank and, in principle, need not be stored). Yet, the fact that these

elements are blank is important; they provide information on which areas of the map sheets are empty although it is inefficient to hold them in an uncompact form. There have been a number of techniques suggested (Rose and Willoughby 1972) to dispense with the need to store this copious information. Generally it is easier to handle the data in their uncompact form, although conceptually and physically it is notably more efficient to adopt a compacted raster format within any sizeable data base.

3.4.1 Data compaction methods

There have been numerous techniques suggested for the efficient handling of sparse matrices within a computer system; some are far more suited to geographic data than others.

To hold a raster file with a bit-map approach using associated look-up tables as suggested by Pooch and Neider (1973; Visvalingam 1976) would provide a seven times reduction in storage overheads for a 8000 x 8000 x 1 byte pixel image which is 98 per cent sparse. In a bit-map scheme, only the non-zero pixels are stored end to end in a one dimensional vector. The positions of these non-zero terms within the two-dimensional raster image are defined by

a 2-D array of set single bit elements; all unset bits in the array correspond to the positions of empty pixels. A scheme of this nature is very appropriate to mathematically-based applications involving sparse matrices such as linear programming techniques and least squares methods in which each element of the array is usually examined in turn (Rose and Willoughby 1972). However, for geographically-based processing the technique does have severe limitations since the retrieval of the non-zero elements is frequently on a random access basis and this will necessitate repetitive and wasteful scanning backwards and forwards to determine the address of each location required. Furthermore, the majority of computer systems available today are byte oriented. A collection of eight bits is the smallest unit of computer storage that can be readily accessed, from most high level languages; bit manipulation is usually possible of course with low level assembly programming.

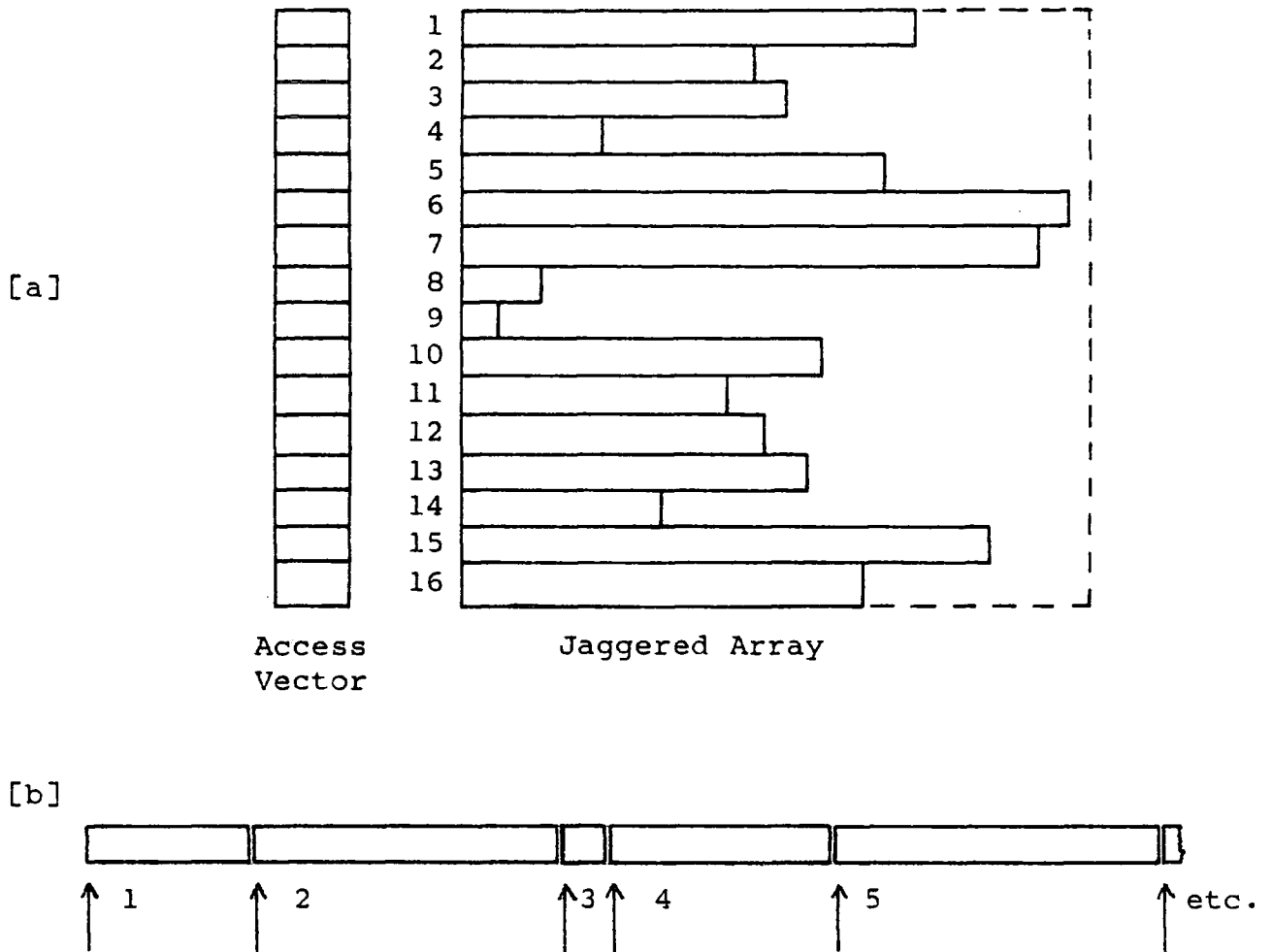
The run-length encoding scheme seems to be far more readily applicable to geographical data processing; several raster-based systems in existence utilise this approach (eg Rhind and Evans, 1980). The ideas of run-length-coding (after Rosenfeld 1969 p 16) are particularly applicable to picture processing especially after the classification of pixel values into a small number of groups. Rosenfeld points out the apparent high probability of specific pixel values

occurring in 'runs'. In this scheme, the start and end addresses of each 'run' are flagged rather than encoding every value in sequence. Clearly no general rule of thumb can be given for the efficiencies of this compaction technique since it is highly dependant on the characteristics of the data. The most appropriate situation will occur where one scan line contains a complete run of identical values. In such an unlikely event it will be possible to store a complete line with two values - the start and end positions. Conversely, the most inappropriate situation will occur in the equally unlikely event of each contiguous pixel in a scan line being coded differently. In such a case no run-lengths can be coded at all and the scan line must remain in its uncompactd form. Undoubtedly any scheme that is chosen must be tuned to suit best the characteristics of the data. For example, in the OS large scales data the longest runs of contiguous equi-coded pixels, excluding the horizontal grid lines, will be the zero terms representing the empty areas. In such circumstances, it will be most appropriate to encode these pixels as run lengths. The reverse approach (i.e. to encode the set pixels as run-lengths) will be more beneficial in smaller scales mapping or more exceptionally in geological and thematic mapping. The degrees of compaction for large scales raster data achievable with a run-length encoding strategy is almost certainly related to the orientation of urban street patterns. For example, in central Edinburgh and

in many American towns, there is a higher degree of streets running approximately east to west and hence such conditions will benefit greatly from raster scan lines running in a similar orientation. Although somewhat arcane, this point does show the need of careful choice of scan line orientation for any sizeable data base. In any event the use of run-length encoding techniques will produce a data structure which, from a programming point of view, is rather more complex than that of the uncompact version. The array which is generated will no longer be regular in the dimensional sense. A jagged array as outlined in figure 3.2a below will result which will require some form of access-vector or look-up table to be incorporated into the retrieval system which supplies the details of the start and end addresses of each scan line, their length in bytes and so on (see Page and Wilson 1978, page 59).

In situations of this nature it will be highly desirable to treat the structure as a single dimensional array within the computer with the access vector giving all associated scan line addresses, and other relevant information. (see figure 3.2b).

Figure 3.2: Indexing with access vectors for jaggered arrays



Treating jaggered array as a single-dimensional array with access vector shown in [a] above providing pointers to the start of each new record (or scan-line).

Experiments have shown that the delta-indexing scheme (Visvalingam 1976) is by far the most appropriate technique for raster-based large scales mapping studied by the author, when such factors as data storage, transmission, uncompression

and general ease of understanding and applicability are concerned. The delta indexing scheme has many similarities to the run-length coding ideas of Rosenfeld: only non-zero items are stored together with their absolute positions in the array. The zero terms (which are highly contiguous in topographic mapping in particular) are stored as run-lengths to represent the difference (or delta) between each non-zero term. In order to distinguish between actual data items and the coded deltas which represent the series of contiguous zero elements, the latter are stored as negative values. This is still possible with one byte per pixel: seven bits enables up to 128 attributes and the eighth (most significant) bit provides for a positive/negative distinction. In this scheme, it is not a time consuming task to regenerate the array in an uncompact form. Using ANSI 1966 FORTRAN on an IBM 370/168, a typical 1:1250 map in compacted form can be expanded into its original 4096 x 4096 pixel raster form in about 20 seconds of CPU time. Indeed, it is possible to scan through each row in the array in a forward or reverse order, since the coded deltas are equally applicable in each direction. Visvalingam (1976) compared the efficiency of the bit-map and coded delta schemes for the 100 per cent population census data for Durham County (see section 3.5.1). Her results are worthy of note and are shown in table 3.3.

Table 3.3 - Comparison of the bit map and coded delta schemas for zero suppression of 100 per cent population data for Durham census county.

(From Visvalingam 1976)

	Prior to zero suppression -----	Bit encoded -----	Coded deltas -----
Size of file in pages	249	121	124
Execution time to read and (where necessary) regenerate file in secs.	11.4	26.2	18.4

N.B. A page in the MTS operating system represents 4096 bytes

This author has performed similar comparisons for an OS sheet and the results are shown in table 3.4.

Clearly the ability to compact an otherwise voluminous data set like a raster image to realistic proportions is an area of considerable importance in the topographic data base industry. The comparisons given above, if typical, would shrink a raster-based national digital topographic data bank from 22,000 magnetic tapes to 1100 (no back-ups included, 6250 bpi tapes with 220,000 OS map sheets assumed). Essentially, however, no single technique can be generally

Table 3.4: Comparison of the bit map and coded delta schemas for zero suppression of OS sheet SO 9283SE held as a 4096 x 4096 raster byte map.

	Prior to zero suppression -----	Bit encoded -----	Coded deltas -----
Size of file in pages	4096	631	208
Execution time to read and (where necessary) regenerate file in secs.	9.3	92.5	21.8

N.B. A page in the MTS operating system represents 4096 bytes
Software was written in FORTRAN IV using an H-extended
optimising compiler on an IBM 370/168

applicable to all types of data; careful system design and tuning are essential to provide an efficient working system with tolerable storage and retrieval rates. In this sense, algorithmic selection of packing procedures may be worthwhile incorporations (as Amidon and Akin 1970 described for vector compaction). In any event, lessons can be learnt from other areas of research which are using a raster-like method of characterising data.

3.5 SOME EXAMPLES OF RASTER-BASED APPLICATIONS OTHER THAN
MAPPING

The technical considerations of two data types which possess raster-type characteristics immediately spring to mind: these are the storage and manipulation of Landsat data and the work of the Census Research Unit at the University of Durham which has successfully stored the whole of the 1971 Great Britain Census Small Area Statistics organised by 1 kilometre grid squares (Visvalingam 1975).

3.5.1 Grid square-based census mapping

Some 149,000 populated 1 kilometre grid squares exist within Great Britain and the total amount of 1971 census data involved was in excess of six hundred million bytes. Moreover, as Visvalingam (1975) pointed out, it is desirable to hold two or even three copies of the data to guard against accidental corruption. Clearly, such large data volumes are costly to store and maintain and a significant tape library is bound to result.

The storage techniques adopted by Visvalingam in her work are worthy of note and can provide the basis for similar management systems of topographic data since grid square

census information is not excessively dissimilar so far as physical structure is concerned and has similar spatial characteristics. The uncompressed 1971 census data for 1 kilometre squares in Britain, as supplied by the Office of Population Censuses and Surveys (OPCS), required at least three double density IBM 3330 disk packs (each with a storage capacity of 200 megabytes). Clearly to store the data on-line in their uncompact form would have been highly inefficient and costly. Additionally, it is considered undesirable to span a data base over two or more volumes wherever possible: the mounting and dismounting of disk packs can be time-consuming and uneconomic in most computer installations. The computer system adopted for the work of the C.R.U. was the N.U.M.A.C. IBM 370/168, a (then) six megabyte virtual memory machine operating under the control of the Michigan Terminal System (MTS). Such a virtual memory environment can manifestly aid a programmer concerned with large data sets since, conceptually at least, it is possible to seize large amounts of fast core; in reality only part of the allocation resides in main memory at any one time and the system automatically swaps partitions of the data or program to and from backing store as required. With large data sets this can induce constant page swapping with very little progress on the job unless careful thought is given to the organisation of the information being handled (Day 1972). MacDougall (1976) has shown how, in handling geographical

data on a virtual memory system, some alternative ways of processing the data can differ enormously in speed of operation.

The basic characteristics of the compaction procedures developed at Durham for the storage of the 1971 census data aggregated by 1 kilometre grid squares involved the suppression of zero elements by the delta indexing scheme (see section 3.4.1) and this accounted for a reduction to between one third and one quarter of the total volume of data. The physical ordering of the data has some importance: it arrived (spasmodically) ordered within 100km squares, each of which contained a variable number of records for the populated 1km squares therein. Since the primary objectives (eg. mapping) involved working with selected variables for all or large proportions of the country, the data were reordered into west to east 1km squares across the country and all of these strips were ordered from north to south. Clearly, to retrieve information for south-eastern England could be a lengthy sequential process since this would involve skipping through many records to reach the desired address in the file. The use of a locational index to provide pointers to the bulk of the data was found to be advantageous. The organisation of the data has specific differences to a raster-based topographic image although these differences are not unduly resolute in suggesting the

C.R.U. data structure to be completely unsuitable for other data types.

Generally, census data can be considered as providing patchy spatial coverage with many (compactable) attributes per area. Map data have complete spatial coverage although they comprise fewer attributes than those held in census data. The compaction of map data is achieved by treating them like census data and removing all cells for which no data are available (e.g. white space in map data or grid squares with a population of twenty five people or less - see Visvalingam 1977). The total area of Great Britain covered by the census is 903,000 square kms, although the total area for which data are available is 149,000 square kms. Visvalingam (1977) has achieved a packing rate of c. six times by virtue of removing grid cells for unpopulated areas on land and sea. In this work, Visvalingam has discovered that indexed sequential data files with a series of index-heading files provides a suitable retrieval system for the data. Her indexing techniques, essentially, adopt a three tier entry: figure 3.5 shows this retrieval process for a specific user-defined location.

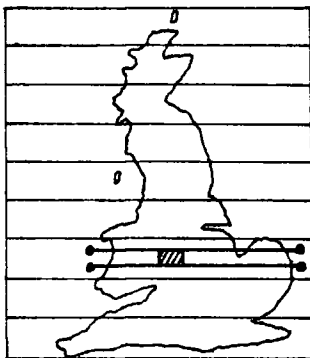
Table 3.5: Levels of index for the CRU data base of 1971

UK population census statistics by 1 kilometre grid square.

Level 1: test if coordinates defined by the user to define required region falls within the UK area.

Level 2: search index file to see what data are available in the required region. Is this adequate for the user's needs ?

Max y



Min y

Level 3: go to scan lines in question via a look-up table to extract required data.

The locational index ideas of Visvalingam and the work of the C.R.U. provides ample evidence of a system involving grid-based data on a large scale which can work quickly and efficiently. In using the system, the methods of data organisation are completely beyond the concern of users who need only access some appropriate subroutine from their own programs, providing a retrieval key (such as a grid reference) and hence providing a working interface with the

census data. Clearly a similar approach will be necessary for any raster-based topographic data structure; the work of the CRU is especially relevant to OS small scales mapping (Landranger series and smaller).

3.5.2 Handling multispectral scanner data

A raster-based data structure operated on a much larger scale is the multispectral scanner remotely sensed data store of the NASA EROS Data Centre at Sioux Falls, South Dakota. An indication of the volume of data involved can be quickly assessed from a summary of the characteristics of the Landsat programme:

- ESSA (Environmental Survey Satellite Administration) and Landsat (formerly the Earth Resources Technology Satellite of the National Aeronautics and Space Administration) scanners are capable of generating 10^{10} - 10^{13} bits of information every day;

- Landsat can produce up to fifty thousand images every

seven days, in which each image contains thirty million bytes, with each image consisting of four channels of 7.5 million one-byte pixels.

- Between March and November 1975, the European Data Centre in Rome (which unlike EROS is merely concerned with European data) generated 700 kilometres of magnetic tape of Landsat data.

Clearly to handle such large data volumes as do Sioux Falls and other data archives requires a carefully designed data structure. A major factor in the design is the range of permissible user requests. Table 3.6 itemises the choices available to a user requiring remotely sensed environmental data in digital form.

Table 3.6: User options when requesting remote-sensed data from the NASA EROS Data Centre at Sioux Falls

1. Area of interest: requests may be specified either in terms of one coordinate pair (latitude and longitude coordinates) which defines the centre of the area of interest or as a rectangular area (by giving the maximum and minimum latitude and longitude of the frames position).

.....continued overleaf.

2. Preferred type of coverage: EROS Data Centre offers four types of data coverage - Landsat, Skylab, NASA-Aircraft or Aerial mapping. For each of these types monochrome, colour or colour infra-red data are available.

3. Preferred time in year of capture: options are available to specify the time in the year at which the data were captured; the year is sub-divided on a quarterly basis.

4. Minimum quality rating acceptable: the quality of remote sensed data can vary dramatically and four options are given to specify the quality acceptable; these range from very poor through poor through fair to good.

5. Maximum cloud cover acceptable: presence of cloud on remote-sensed data can seriously affect their uses; five options are given to specify the maximum amount of tolerable cloud cover: 10%, 30%, 50%, 70%, 90%.

Clearly, to offer a user so many options reflects an efficient cataloguing system. All the raster data held by the centre are stored on high density digital tapes (HDDT) which have a packing density of 10,000 bits per inch. As soon as a user's request for data is received, a query is made to

an on-line catalogue management system in an attempt to match up the user requirements to those data which are actually available. The system is somewhat more advanced than the C.R.U. data structure discussed in section 3.5.1 since the data available or suitable alternatives can be listed and sent to the user who can select the most appropriate images available to suit his needs. These data are then retrieved from the HDDT's and uncompactd from the internal EROS data formats on to computer compatible tapes (CCT), typically with a packing density of 1600 bits per inch, which the user can decipher on his own installation. Harris (1979) gives an appraisal of the formats of Landsat CCT formats.

The storage of the 1971 U.K. census data and remote sensed multispectral scanner data are two examples of raster-based spatial information other than topographic data with which the author has had personal experience. The two are highly dissimilar in terms of data content and volume: nonetheless they provide good examples of raster data held both on-line and off-line in a working environment serving user needs. To hold complete coverage of the U.K. large scales topographic data in a raster-format would be no larger than the quantities handled by the EROS Data Centre as one part only of their business. Thus, it is technically feasible to hold topographic data in such volumes and a precedent exists for the logistical operations needed to run a service based on

these data. Nonetheless, the applicability and usefulness of storing topographic data in raster mode, compacted or otherwise, is still an area of much theoretical debate and some consideration is given to this in chapter four.

3.6 THE REVISION OF A RASTER-BASED TOPOGRAPHIC DATA STRUCTURE

A fundamental factor governing the applicability of any topographic data structure must include provision for the update of information held in the archive. The large scales Ordnance Survey data are continually in a state of transition, largely as a result of the department's policy on resurvey. Since the implementation of the Davidson Committee report of 1938, OS has adopted a system of continuous revision. It is now (in theory) possible to obtain maps of all parts of the UK which are out of date by no more than fifty units of change (each normally equivalent to one new house) and often to a level much better than that (see Simmonds 1978). In the foreseeable future, OS anticipate that the update of the digital files will operate on a similar basis to the present graphical procedures (OS 1981b).

It seems likely that remotely sensed data will assist in the revision of a raster-based topographic data archive although this has not been tried on production systems at

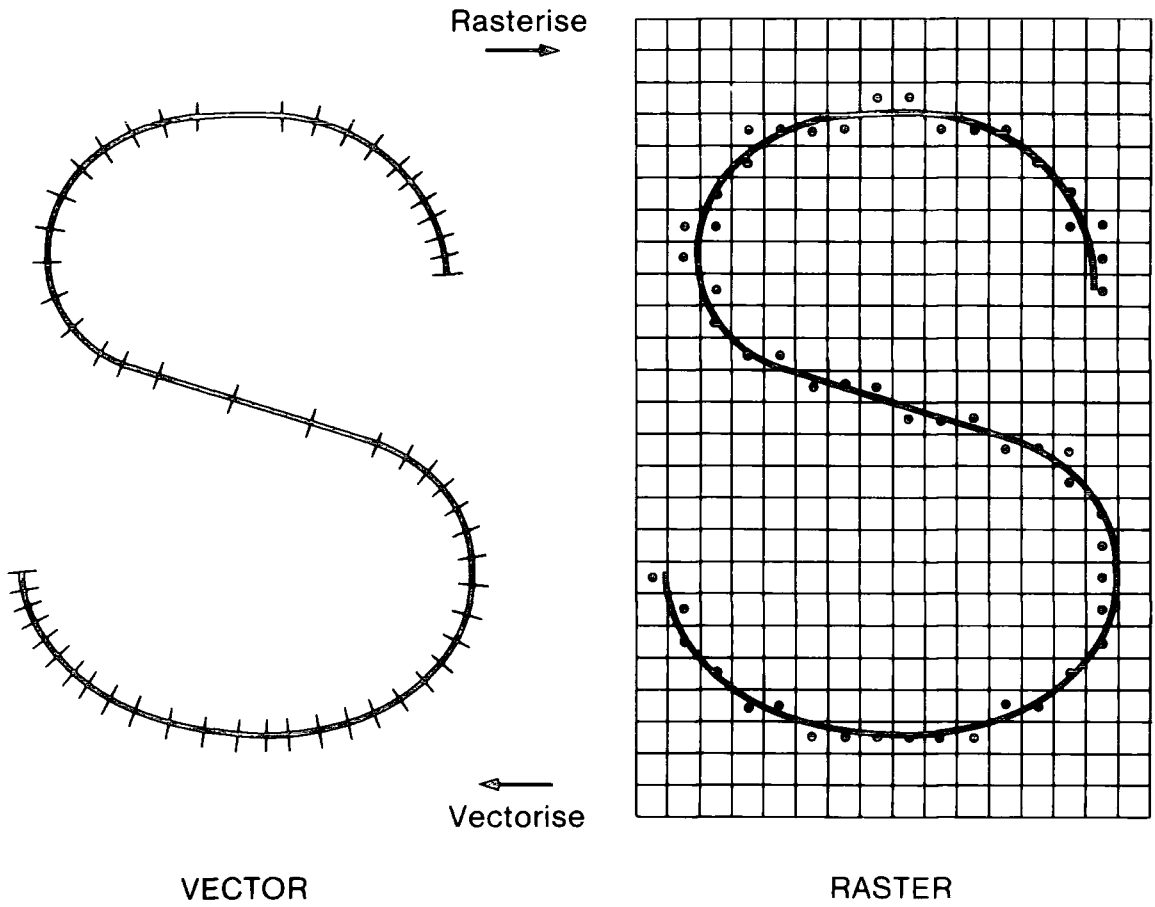
present. Any satellite-housed scanner with a spatial resolution of 25 metres at best and 80 metres as standard (such as the Landsat programme) is hardly of interest when considering the storage of topographic raster data compiled from basic scale maps and typically with spatial resolutions of ten centimetres. Moreover, an operational survey cannot base a service on systems with irregular data supply or on experimental facilities. At present, it seems likely that, by the mid-nineties, various techniques may offer products which allow regular updating of maps of the Landranger series (1:50,000) or of smaller scales, although here too it seems highly likely that any interpretation methods for remote-sensed data will still be centred around visual inspection, perhaps in conjunction with computer assistance. Thus, Ordnance Survey sees remote-sensing techniques offering a supporting role for intelligence gathering and small scale map revision (Thompson and Shreeve 1978). Clearly, prior existence of a long-established system for detecting and incorporating change plus the vital need for a regular and cost-effective acquisition of information makes it highly unlikely that remote sensing will provide more than a supplement to existing methods.

Though the development of a totally automated revision facility for topographic data is highly unlikely for several decades yet, the revision of raster-based data is, in

general, far less involved than any vector equivalent: the complexity of a cellular data archive for topographic data is simpler than the vector strategies described in chapter two. To amend, insert or delete a feature in raster mode is a straightforward substitution of pixels whereas to manipulate vector strings with their associated cross-referencing to other features is far more involved. Chapter four concentrates more closely on the differences between raster and vector notations.

3.7 RASTER DATA HANDLING APPROPRIATE TO MAP DATA

Thus far, a quick resumé has been given on basic raster definitions with the help of appropriate examples and applications. For several years now researchers in the field, chiefly Americans and Europeans, have suggested that the manipulation of map data in raster form is as straightforward and often easier than is the handling of vector quantities, although conceptually the reverse is more likely to be expected. The use of raster methods for handling map data will now be discussed. Figure 3.7 shows an identical feature in both raster and vector mode. To convert from (a) to (b) will involve a rasterising process and to convert from (b) to (a) will be a vectorising process.

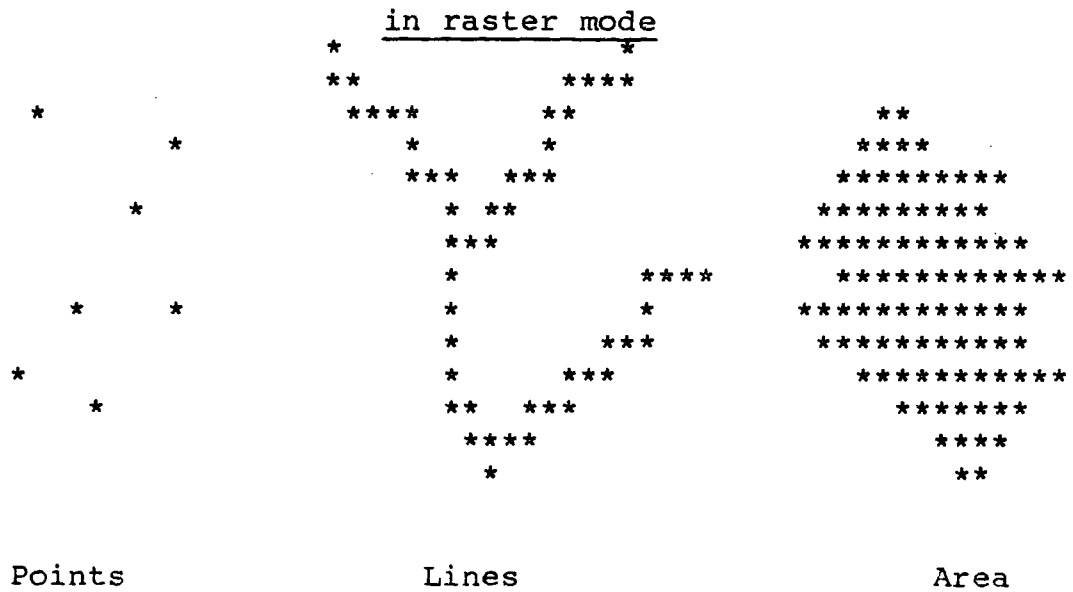


THE DISTINCTION BETWEEN VECTOR AND RASTER NOTATION

Figure 3.7

The use of raster methods to define points, lines and areas is as amenable as is a vector notation; indeed, the representation of areal data is especially suited to a cellular strategy. Figure 3.8 below shows an example of the three cases of points, lines and areas and suggests that the representation of areas by raster is more appropriate than vector polygonal data as was outlined in chapter two.

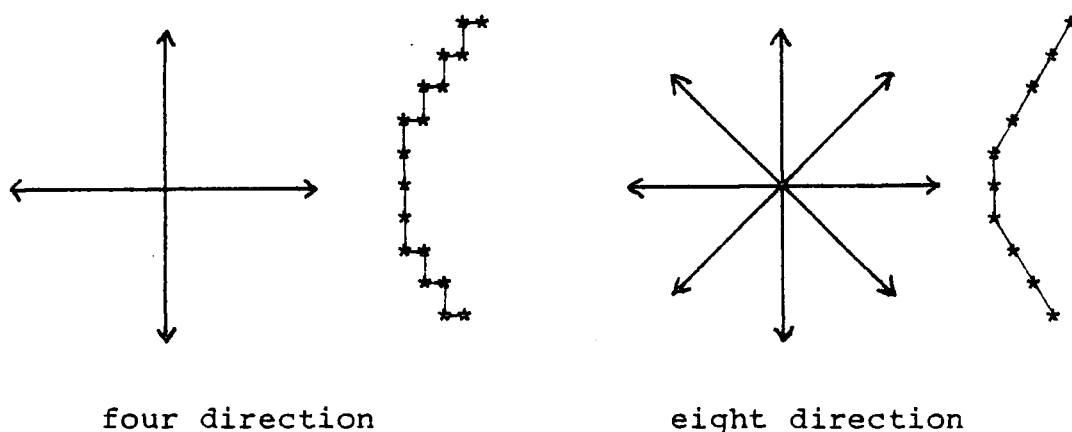
Figure 3.8: The representation of points, lines and areas



As with a vector approach, it is important to consider and itemise a series of rules governing the conventions for the definition of linear features in raster mode. A typical example is the definition of pixel connections in either four or eight directions; the latter will give a more accurate representation. Figure 3.9 shows the effects of a four and

eight direction pixel connection for a linear feature. The convention adopted is important in the design of manipulation software - in particular the raster-to-vector conversion process.

Figure 3.9: The effects of four and eight direction pixel connection



Within specific tolerances, which can be defined by the user, raster data can be made to be as applicable to map data as are strings of vectors, albeit with different algorithm design and software tasks. Problems which arise lie chiefly at the data capture stage: section 3.3 describes techniques for the graphic-to-raster conversion of images and chapter seven gives an overview of commercially available hardware to perform the task. Little or no explicit definition of real-world entities and of their connectivity exists within the data emanating from the raster scanners; in short, they

provide a raster equivalent of vector 'spaghetti' files (Peucker and Chrisman 1975). It is the ability to decipher these 'spaghetti' files to introduce some data classification and topology which is fundamentally difficult at the present time. Chapter six reports on some work by the author which attempts to introduce some 'meaning' into otherwise 'spaghetti' data using a battery of array processing techniques.

Most attempts to decipher scanned data have relied on the partial intelligence offered by grey levels. Today most advanced scanners (see chapter seven) have the ability to detect both pixel feature position and colour (or grey level) although the latter can vary from image to image. Grey level thresholds have normally to be set by a human operator to suit the characteristics of the source document (see section 7.2.1). No known scanner has the ability to detect and classify a whole feature by some unique signature; this must be carried out as a post-hoc stage utilising colour, line thickness and (usually) large amounts of human interaction. Clearly such an approach can be lengthy, costly and have a large percentage of errors. The early use of colour scanners has been restricted to the production of colour separation plates for printing purposes: for example, eight colour maps can be scanned to produce digitally three process colour plates and black for other printing specifications. Any

attempts to classify feature types automatically have largely emanated from the work of image analysts: the work of Weber (1980) and the Image Analysis Group of Oxford University (Black et al 1979) are worthy of note.

3.7.1 Basic image analysis for feature recognition

The process of skeletonising or thinning lines represented by a swathe of several pixels wide to a single pixel wide centre line can be used as a technique for classifying feature types. The number of pixels required to represent the totality of the width of a linear feature on a map is dependent on the line weight of the map line-work and the chosen resolution of the pixels.

Scanned linework can vary from between one or two pixels in width to as many as a dozen or more; indeed, with poor quality linework or with pecked lines (in which the pecks are of irregular length if drawn by hand), they may even disappear completely in places. It is the job of the human operator to set the threshold grey values to some appropriate level to ensure sufficient digital representation of all required information on the source document to be scanned with no recording of extraneous 'noise'. A general investigation of line thinning techniques by the author has

suggested that no robust algorithm exists. The diversity of techniques have been developed largely by research groups or commercial concerns interested in this work.

Since most raster data sets are voluminous, it is advantageous to approach the line thinning problem with as few passes as possible through the data. It has been pointed out by Harris et al (1981) how line thinning algorithms place restrictions on the number of pixels allowable to represent a line. This (they claim) is chiefly to reduce the processing time necessary to perform the line thinning process. Most thinning techniques seem to have taken either a mid-point algorithm approach or one which 'nibbles' away at each side of a line until one pixel is left. Good descriptions of some working systems have been given by Boyle (1979) and Harris et al (1981), although there are several others: most manufacturers of raster scanners can provide software, often operating in real time, to thin the scanner data although (not surprisingly) they are reluctant to publish descriptions of their algorithms (see chapter seven). Most techniques adopted thus far have been designed for sequential-type processors, often for mini-computers with a strictly limited amount of available fast core and disk space. The majority of raster scanners, generally, require connection to a controlling mini-computer for operation. An interesting alternative solution has been offered by ICL (Hunt pers comm

1980) which adopts a parallel algorithm to run on the ICL Distributed Array Processor (see chapter six). The Research and Development team at ICL are developing an efficient line thinning algorithm to facilitate easy and precise automated hand-written character recognition for improved user-interface with computers. The system will be based on raster scanning ideas akin to those in use in cartography. Thus, local raster scans convert the handwritten characters into a digital form then, after subsequent rapid line thinning procedures operating on the parallel processor, attempts are made to match the characters (in pixel form) with a pattern library of alphanumeric characters held as bit patterns. The problems involved should be less than the topographic data examples because there is a finite number of character shapes possible. ICL predict a high rate of success in recognising characters in this way; a description of the methodology adopted is given in British Patent specification (serial number 1345032 - dated 1974).

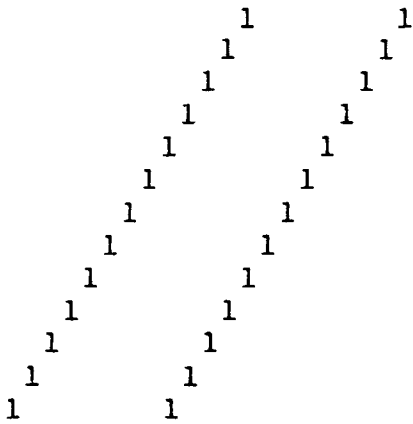
In conclusion, line thinning is a necessary pre-requisite to any further data processing although no complete and robust solution exists. There have been several techniques suggested by various research groups although all have some minor idiosyncrasies in their results - see chapter seven. Apart from reducing the characteristics of the raster data to more manageable volumes, the process of line thinning (or

skeletonising) in itself is a powerful tool for image analysis and feature classification. Simple feature classifications can be performed automatically where the thickness of line work is known to differ systematically between features within the same image. With appropriate software, it is possible to thin some lines to the extent that they are removed completely, thereby leaving only the thicker lines in the raster image. In this way it is possible to code the thicker lines in isolation from the thinner lines. The thin lines can be isolated too; by subtracting the thick lines in isolation from the original image, only the thin lines will remain. A cycle of this nature can be performed several times; however, the amount of feature classification possible by this method will be dependant on the variability of the scanned data. Such an approach would be unsuitable for most mapping produced by Ordnance Survey. The cartography specification for the Landranger (1:50,000) series, for example, quotes the line weight to range from 0.15 millimetres for most features up to 0.6 millimetres. Since the majority of features utilise the narrower gauge of line, then very little feature differentiation is possible on line weight alone, although features could probably be isolated using a combination of line weight and colour. The line width characteristics of the larger scale plans are very similar, and since they are monochrome then no assistance from colours is possible. Many

of the linear patterns on the larger scale plans are represented by dual parallel lines: roads, railways and footpaths are examples. The detection of parallel lines in a raster mode, especially on a serial machine, although difficult, is possible according to Weber (1980) using distance transformation methods. He suggests quite successful feature classification to be possible by this technique, especially when allied to line thickness of the map features. Rosenfeld (1969, p 141-147) describes the nature of distance transformation techniques in the recognition process of features in a digital image. A simple example is shown in figure 3.10.

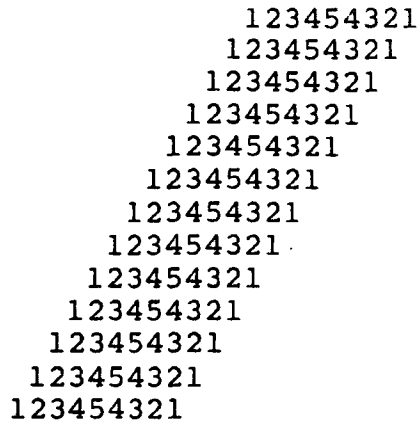
Taking the original bit-map as shown in figure 3.10a, a distance transformation technique is used to assign pixels bounded by the parallel lines. The values assigned to each pixel are a measure of the shortest distance to the nearest bounding line. The examination, say, of the pixels with values 2 or 3 give indication of the course of the feature and its thickness. This is repeated several times for different pixel values until a unique signature is determined for the feature under scrutiny. Software to perform these analyses is either complex or inefficient on a conventional sequential computer but is much easier to produce on a cellular array processor such as the Goodyear STARAN or the ICL Distributed Array Processor (see chapter six).

Figure 3.10: An example of a simple distance transformation to recognise a road carriageway on an OS 1:1250 map sheet



(a)

Original feature in
raster



(b)

Distance transformation
of original feature

Essentially, then, the ideas suggested by Weber (1980) are logical extensions of the image processing ideas of Rosenfeld (1969). The adaptation of distance transformations is useful for many sorts of pattern recognition task. Clusters of pixels which are considered to comprise a pattern of known characteristics can be simplified into some unique skeleton for easier comparison with a pattern library. This "pattern matching" strategy is a well-established technique used in many image processing applications. A simple example of a major road/minor road junction is shown in figure 3.11.

Figure 3.11: The use of distance transformations to detect a road junction in the OS 1:1250 scale data.

(after Weber 1980, figure 14)

654433221 12211		
554332211 1221 1		22
54432211 11211 1		2
4433211 11211 11		2
333221 11221 112		22
222211 12211 122		22
11111 11211 1111		2
11221		22
1111112111111111	111 2	11111
1221		22
11 11211 1111111		2
1 11211 1122222		2
11221 112233333		22
12211 122334444		22
11211 1123344555		2
2221 11223445566		

Distance transformation
of the raster image

Skeletonise the
image to detect the feature

3.8 USER EXPERIENCE

From a U.K. point of view, and to a certain extent in the U.S.A. too, digital cartography and remote sensing have until recently been virtually separate subjects. At present there is very little user experience in Britain associated with a raster-based organisation of cartographic data but see Diello

et al (1969) for early US examples of raster strategies in cartography. Generally, it is the developments of other disciplines which have gradually infiltrated into the cartographic fraternity and accentuated any real effort to adopt a raster approach. Moreover, computer cartography has tended to be dominated by the problems of digitising and the subsequent production of acceptable graphical output of data. This concentration on both the input and output has to some degree diverted attention from the third of a set of closely related problems: the appropriate organisation of an efficient cartographic data structure. Arguably the characteristics of the input stage and desired output formats are closely allied with the internal storage and utilisation of the data. Rhind (1976) has strongly rejected these philosophies claiming that input, storage and output need not be connected.

It was the American textile printing industry which, in the late 1960's, stimulated the development of a raster scanner with the ability to detect six colours: the company commissioned for an initial project was Scientific Technology Ltd. (now Scitex Corporation), now probably the world leader in the field of raster scanning hardware (see chapter seven). By 1972 hardware and appropriate software for a controlling computer was made generally available by Scitex Corp. albeit with rather limited facilities. It was in 1975 when the

Scitex Response 200 system was introduced; this became the first successful raster-scan system in the textile printing and decorative printing industry. There are now in excess of one hundred of these systems in agencies all over the world (Scitex Corp. pers comm 1981). Since the late seventies, applications groups have begun to find common links between this early technology and cartographic work: the introduction of a modified cartographic scanning system (the Response 250) is an example of these close links. There are, however, other systems available with similar characteristics; these are reported in chapter seven.

3.8.1 A United States view

The work of the U.S. Army Engineer Topographic Laboratories (ETL) at Fort Belvoir, Virginia is worthy of note; they have gained experience in both vector and raster formats (Babcock 1978). The implementation of an experimental drum scanner/plotter with variable scanning resolutions has provided a source of raster data. They claim that the raster hardware offers advantages in speed and accuracy over their linear-oriented systems although the processing of such data is proving to be a major problem. Babcock (1978) points out the sheer volume of data emanating from the scanner. He quotes the total number of points

generated for a graphic 24 x 30 inches to vary between 720×10^6 to 45×10^6 , depending upon the resolution chosen. Using standard run length encoding techniques, it has been possible to reduce these volumes by a factor of seven.

It was at the ETL where the first attempts to use array processing techniques were carried out. In an attempt to reduce the large amounts of 'mill-time' being necessary for data manipulation on a conventional serial computer, software was developed to run on the Goodyear Aerospace STARAN associative array processor. More detailed consideration of these specialised parallel processing techniques is given in chapter six.

A number of related research projects undertaken by ETL which have examined techniques for the efficient handling of large volumes of geographic data can be commended here. Their approach to the efficient handling of terrain elevation data is worthy of special mention. Such data may be generated in linear or raster form depending whether a conventional digitiser or a raster scanner is used. However, data stored as digitised contours, while most efficient for archiving purposes, cannot readily accommodate user needs other than the recreation of contours. The most actively used format for terrain elevation data in the United States is a raster-based format; a matrix of spot elevations (with

resolution typically one point per 0.01 inch at 1:50,000 scale) can expedite contour interpolation, shaded relief patterns, slope gradient and azimuth calculations, determination of intervisibility and a number of other related products. In excess of three million points are required for this format and E.T.L. investigations have suggested that careful modelling techniques adopting low-order polynomial coefficients centred over selected grid points can produce a compaction of up to 80:1. These results, allied with advanced optical digital encoding of data onto video digital discs, suggest that well over six thousand terrain maps at 1:50,000 could be stored on a single 12-inch diameter volume (Babcock 1978).

In the Geography Programme of the United States Geological Survey, land use mapping has been attempted with both polygon and cellular formats. Mitchell et al (1977) itemise the advantages of cell-based data for several manipulative tasks: later versions of GIRAS (a Geographic Information Retrieval and Analysis System for handling land-use and land cover data) have been modified to accept grid cells as input and also to convert polygon structures to a raster format for various analyses. Furthermore, the Geography Programme has investigated the use of a raster scanning device (an I/O Metrics Corporation SWEEPNIK laser scanner) in a 'line-following' mode in an attempt to speed digitising rates

and reduce many of the problems inherent in manual procedures. Unfortunately, Mitchell et al (1977) found several disadvantages with this approach; weak lines on the original can be missed and thicker lines can merge together. However, the researchers claimed that such a system could be error-free and cost-effective with source documents of high quality (e.g. scribed line-work providing good contrast with the background).

3.8.2 The European view

With the exception of image analysis groups, research into raster-based mapping in Europe has tended to be primarily concerned with map-production and associated techniques of converting raster spaghetti into vector strings. The work of IGN Brussels (Loodts 1981), IFAG Frankfurt (Weber 1980) and IAG Oxford (Harris et al 1981) are examples of European groups interested in this field. Most of these research workers, with the exception of Weber, suggest the rapid conversion to vectors for manipulation and storage to be the best philosophy. This author has detected no evidence of European research into a long term, raster-based data structure for cartographic information. All European production systems using raster methods are chiefly concerned with providing more cost effective data capture and edit

facilities for the creation of further, modified graphics rather than a raster-structured data base useable for a variety of tasks.

Kummerly and Frey AG, a much respected Swiss map producer, have found the partial use of raster methods to be cost effective in the creation and revision of road maps (Kummerley and Frey pers comm 1981). Their flowline, although rather simplistic, serves as a good example of the use of Scitex hardware in production: they operate a scanner, a raster-driven laser plotter and two raster-driven interactive edit/work stations. Essentially the revision exercises have involved the scanning of up to twenty 'astrofoil' map sheets in which there are thirty-eight overlays per sheet. Since the format of each map is larger than the dimensions of the scanner, then the data capture has to be made in two passes with a reliance on the accuracy of pin-registration marks. Once in pixel form, any map revision has proved to be easy and efficient. The maps are multi-coloured and well suited to the Scitex facilities. Symbol placement and minor modifications are easy to achieve in the raster data with the interactive work station.

Kummerly and Frey AG also report that the production of new road maps adopting a raster strategy is proving to be cost effective (Kummerley and Frey pers comm 1981). Existing

data are being scanned and digitally reduced and transformed to an appropriate projection for replotting. This provides the cartographers with a compilation sheet to be redrawn by hand with any modifications where necessary. Symbols are compiled on a separate sheet; once rescanned, they are merged with the line-work interactively on the edit station. A faster turnaround of production is claimed although text placement still relies on the raster scanning of letters which have been 'stuck' onto a sheet of astrofoil.

The simplistic uses of raster methods in this way are far removed from the advanced applications as suggested by academics such as Boyle, Peuquet and others, but nonetheless they do provide an example of a pilot production flowline adopting fairly controversial techniques which is reported to be more cost effective than conventional vector-based digital methods. The users of Kummerly and Frey AG road maps have, at present, no interest in these new methods of production and the map makers themselves are not anticipating uses of their data other than map production, nevertheless the status quo is sure to change sooner rather than later as technology in image analysis and improved mass storage devices reach maturity.

3.9 CONCLUSIONS

This chapter has attempted to introduce the concepts of raster technology with respect to cartographic applications. In principle, a grid-based method of storage and manipulation of environmental data can be as appropriate to mapping tasks as the more conventional vector methodologies. However, with the exception of one or two research groups and a few commercially available systems, there are few working systems from which to draw positive conclusions on performance and efficiencies. Clearly, software design and choice of hardware in a raster-based mapping system must be carefully geared to suit the types of manipulative tasks and output characteristics required.

Nonetheless, raster technologies have a future role in mapping systems, although there is a need for careful advancement of present techniques and knowledge. It is anticipated that the data base management of such data will be unlikely to present many problems if controlled compaction and indexing techniques are adopted. Major problems will arise not so much because of the considerable bulk of the data sets but in the efforts to convert the hard-copy map sheets into a digital representation with adequate content and structure for future (unknown) purposes. Such problems are not, of course, restricted to raster techniques; similar

dilemmas are already facing the vector-based methods of data capture. There is chiefly a need for research and applied effort to develop the post-hoc validation and refinement processes for the output from the raster scanners. The devices now have more than adequate detection specifications for most tasks. The next decade must see much improved image analysis methods by both automatic and semi-automatic means.

4 COMPARISONS OF RASTER AND VECTOR REPRESENTATIONS

4.1 INTRODUCTION

The distinction between raster and vector-based representations for geographic data has now been made: chapter two concentrates on descriptions and definitions associated with vector notations with specific examples drawn from working systems and chapter three provides similar considerations for raster methods. It has been suggested that many applications are more suited to one notation or the other although very little consideration in the past seems to have been given to itemising which representation is best suited to which function. Such a deficiency in attention by research workers has arisen, most probably, because of an absence of any specification of tasks appropriate for a national topographic digital data base to meet. It is the intention of this chapter to consider the suitability of the two techniques for a number of defined applications. Attention will also be given to such factors as revision of the data base and the quality and efficiency of data capture and presentation.

No data base system for topographic information is known to the author which handles vector and raster-based formats and cross-references between each on a routine basis. All systems have taken on one methodology or the other with a clear distinction between the two, although a number of vector systems may use a 'gridding' technique to compare data sets. Apart from this, the uses of vector representations seem to be rather more abundant than are those of grid-based systems. This predicament has arisen by the domination of the problems of digitising in computer cartography. It is natural for cartographers to find similarities between map drawing and manual digitising methods and hence to be prejudiced in favour of holding information in a linear fashion on a feature by feature basis. The early devices specifically designed for data entry and display lent themselves readily to this method of data organisation: the manual digitising tables (see chapter two) and the vector plotters are notable examples. Moreover, by far the majority of commonly known cartographic algorithms are vector-oriented (e.g. Douglas no-date), thus very little implementation work has been necessary to develop quite acceptable and useful production systems. The work of Ordnance Survey over the last decade demonstrates convincingly how this policy is quite adequate for the storage of map data intended purely for the generation of graphic images on demand at some later date. Increasingly, however, the need for relational links between

topography and other environmental sciences is becoming obvious. It is for this reason, in particular, that the reformatting and detailed manipulation of data in a raster format is desirable to facilitate easier links to other data sets. The interface between map and remote-sensed data serves as a useful example. For many years digital cartography and remote sensing have been virtually separate subjects: this is evident with the establishment of individual societies for the two in Britain - the British Cartographic Society and the Remote Sensing Society have very few links. There seems little reason for such clear distinctions between two disciplines which are both concerned with capture and management of environmental information. The fact that members of one tend to produce digital data in vector form and those in the other produce data in raster form has probably accentuated this unreal dichotomy. Present indications of the increasing resolution of pixel data to be gained from the next generation of satellites (see Doyle 1978) give more impetus to the collaboration between remote-sensing and topographic mapping and to the desirability of being able to move fluently in both directions between raster and vector data sets.

.2 THE BASIC CHARACTERISTICS OF DATA HANDLING

It has been suggested by Peuquet (1979a), Weber (1980), Davey et al (1977) and others that raster methods of spatial data handling have as much potential as any vector equivalents. This chapter examines some of the effects that these two approaches have on the many sections in a topographic information system.

.2.1 Data entry

Chapters two and three have outlined the processes involved in converting a map sheet into vector and raster formats respectively. To transfer a map sheet into a pixel-based structure automatically will require access to one of the many scanning devices (see chapter seven). Automated vector capture seems less widespread; there are far fewer examples of commercially available line following devices: the Laser-scan Laboratory's Fastrak (see chapter two, Howman and Woodsford 1978, Rhind 1979) is the only device known to the author which has proven to be successful in a production environment, but see Wohlmüt (1976) for the description of a similar device manufactured by i/o Metrics. Vector methods have, of course, long been well-established with the use of manual x,y digitising tables; there is no

raster-based counterpart to these and so any comparisons between the two methods should be restricted to the technology-intensive automatic methods.

Clearly the time required to digitise a map sheet is crucial in assessing the throughput rates of a data entry system. Both automatic line following and raster scanning offer dramatic improvements over manual methods for some data (see section 2.5.2): it is difficult, however, to compare effectively the speeds of operation for the two types of system since both have a different collection of stages. Current acceptance tests being undertaken on the Fastrak in the Ordnance Survey headquarters indicate that typical 1:1250 plans in the dense West Midlands area can be digitised in between four and twenty hours; this is only a marginal reduction in the digitising rate of the conventional manual flowline described in chapter two. Further, the provisional results being obtained from the Fastrak still comprise many errors and require large (presently unknown) amounts of data editing on the IGES interactive edit system implemented by OS. In practice, it is difficult to assess clearly the performance rates of a Fastrak with production work. Experience gained from the installation in use at the Mapping and Charting Establishment (MCE) has shown that the speed of digitising varies enormously for differing contour data; line curvature, density and image quality are all important in

assessing throughput rates (Howman pers comm 1982). Generally, crude assessments of the speed of digitising for contour data by the Fastrak at MCE are quoted as being ten times faster than a manual digitising process (Howman pers comm 1982).

The initial set-up times for both Fastrak and the manual digitising tables at OS seem to be of a similar orders (c. 15 - 20 minutes), though the use of a Fastrak method requires the calibration of more parameters such as line width thresholds and phase detection. However, it is the nature of the data being digitised which is slowing the Fastrak down from its potentially high line-following rates. The initial tests show it to be faster to digitise manually many features than to follow linear features automatically, then validate the results, filter, smooth and so on. These provisional results (although limited in extent) are disappointing for production requirements. In February 1982, the system required careful modification and extensive tuning to be suitable for the complexities of the large scales data capture. It is hoped that modifications to the Laser-aid software (Laser-scan Laboratories 1981a) can improve upon these provisional figures.

Undoubtedly, automatic digitising by the Fastrak has the advantage of being performed under operator control - any

associated feature-coding and checking being included where necessary. The digitising time on a raster scanning device is only a pre-requisite to much more lengthy post-hoc validation, feature-coding and general tidying-up. It is highly likely that the Fastrak system will (in time) provide relatively clean data suitable for immediate entry into a data bank with the need for only limited correction. The effects of decision-making and assistance by a trained operator throughout the capture process, with the inclusion of attribute codes and other information, is expected to be instrumental in ensuring information for immediate archiving. However, at present, larger amounts of operator assistance and intervention are necessary than is considered desirable for a cost-effective production system. The need for continual alertness on the part of the operator, who must not only respond to queries made by the system but also interrupt the line following process when errors in the selected route (frequently) occur, suggests a highly demanding job both physically and mentally. The automatic recognition of junctions is still inadequate at this early stage and Laser-scan Laboratories are attempting intense development into this option at the present time (Woodsford, pers comm 1981). Nonetheless, the feature coding of the line work is expected to be less problematic by a Fastrak, than with a raster approach since the operator has easy-to-use programmable function buttons at his disposal for rapid real time

intervention.

Understandably, the throughput rates of digitising with a Fastrak-like machine are entirely dependant on the complexity and volume of the data being digitised. This is not the case in raster scanning in which the same time is taken for map sheets of all complexities although size of the plot and resolution of the scan lines will be relevant. Indeed some of the modern flat-bed scanners (see chapter seven) have the ability to speed up and slow down as data complexities vary, although this should not have such a marked effect on throughput. Bench-mark tests carried out by the author with the kind cooperation of various raster sites throughout the world are discussed in chapter seven. A West Midland's sheet at 1:1250 scale was scanned on a Scitex Response 250 system at the United States Geological Survey in Reston, Virginia with a pixel resolution of twenty lines per millimetre. The scanning time was reported to be seventy-two minutes with a further fifteen minutes required to set-up the machine and prepare it for the job.

These more desirable rates of digitising cannot be compared explicitly with vector methods since a raster strategy has much different characteristics of operation. In particular, efficient operation of a raster-scan device can only be achieved if it is uninterrupted as scanning is taking

place. Operator control can only be provided at the set-up stage when such parameters as pixel resolution, detection thresholds and scan dimensions can be assigned. The result is a digital raster image which requires further manipulation before it can be compared with the results from the Fastrak. Any data validation and sorting of the disordered pixels is carried-out post-hoc. The extent of operator intervention required in this later manipulation stage is unknown at present, although it will be a lengthy process for the complex 1:1250 and 1:2500 mapping.

The three stages in a raster-driven data entry system after scanning are: the assignment of a feature coding scheme, line thinning and vectorising. The ability to convert from raster to vector notation is difficult to achieve efficiently by computer and, although one or two are beginning to show promise, no appropriate solution exists at present. With little or no detailed experience in this post hoc handling of OS raster files, it is difficult to assess the problems and characteristics of the techniques involved. One must rely on the reports from similar research with other map data; a number of research workers have shown these stages to be lengthy processes (Harris et al 1981, Peuquet 1981). Harris et al (1981) cite a 3000 x 5000 pixel image of a polygon map as an example: 281 cpu seconds on a DEC 10 are required for line thinning and 268 cpu seconds for node and

segment identification (there were 3600 segments and 2260 nodes). Clearly these are sizeable process jobs which will certainly reflect in the figures of elapsed time, especially in a multi-user computing environment. It is difficult to provide any real comparisons for an OS map sheet since no tests have been carried out to reproduce results similar to those being obtained by the Fastrak. A raster image of an OS sheet was processed post hoc with vectorising software written at the Image Analysis Group at Oxford University (see Harris et al 1981) on behalf of the author in order to assess the feasibility of such an approach for serious map production. The software introduces some topology into the data, by generating vector strings held as link and node data to facilitate map-feature reconstruction as polygon boundaries. The DEC 10 computer at Oxford required 1293 cpu seconds for line thinning, a further 992 cpu seconds for node and segment identification (there were 9385 segments and 11357 nodes detected), and 100 cpu seconds for post hoc filtering and line smoothing. Clearly these processing times are lengthy and are a good indication of the computer access time required for an 8000 x 8000 raster image which is to be expected for OS large scales data. An elapsed time of three to four hours would be typical for any mainframe performing this task.

To consider the two systems (line-following and raster scanning) by speed of capture alone suggests very similar throughput rates to be possible. Indeed, from an Ordnance Survey point of view, it is envisaged that Fastrak will be faster since the results from the raster scanners, even after any post-processing by some Oxford-like software would further require considerable time to be spent on an interactive work station for data validation and feature code assignment. The complexity of a map sheet is unlikely to alter scanning times but will have a considerable effect on post-processing rates in a similar fashion to its effect on the line-following process. More detailed consideration of the quality of the raster scanning results are given in chapter seven. No tests have been made on the ability to attach feature codes to raster-captured OS data although this is expected to be operator-intensive. Such a project would be highly desirable to assess the full capabilities of raster-based systems. Nevertheless, on the basis of visits to seminars organised by two leading raster scanning manufacturers, infinitive assessments can be made of many of the problems involved and these are considered below.

4.2.1.1 The quality of the source document

Both approaches to digitising are highly dependant on the image quality of the source materials: the Fastrak requires photographically reduced microfiches, typically 98 x 68 mm, with a high contrast of linework. The source documents for scanners must vary to suit each model: some utilise photographically reduced microfiches whilst others can accept full size plans on film or paper. All require to be of high quality with a clear contrast between the line work and the background. The quality of some Master Survey Documents (MSD) (see Marles 1971, Adams 1979, OS 1981b section 4.2) will prove to be unsuitable for any raster scanning device: a situation which is likely to arise with the Fastrak system too. Certainly provisional tests have shown this to be the case: currently only the highest quality image is suitable. It will be necessary for OS to enhance the quality of many of the source documents before mass digitisation can begin. The quality and durability of many of the MSD's comprising the national survey archive in Britain is inappropriate for any photographic enhancement (Smith pers comm 1982). There are, for example, currently c.11,000 MSD's which display the map line-work in blue ink (the result of early OS policy to hold the survey record in a form which cannot be photo-copied). In such instances, it is impossible to manipulate the images photographically for modification for automatic scanners or

digitisers. No assessment of the overall image quality of the OS survey archive is available, although studies of this nature are currently being undertaken by the OS; it is envisaged that between five and twenty five per cent of the total number of basic scale MSD's are unsuitable for any scanning process or, indeed, any photographic enhancement (Wesley pers comm 1982).

The effect of any preparation stages will have importance in assessing long term throughput rates. Clearly much costly cartographic redrawing will be necessary in many cases at the outset, disregarding whether a line-following or raster scanning strategy is adopted. Early assessments (from discussions with OS personnel and with scanning manufacturers) seem to suggest that both automatic vector and raster devices require very similar preparation stages. In essence, it is more likely to be the ease of capture which has importance in assessing the two methods.

4.2.1.2 Suggested raster-based work methods

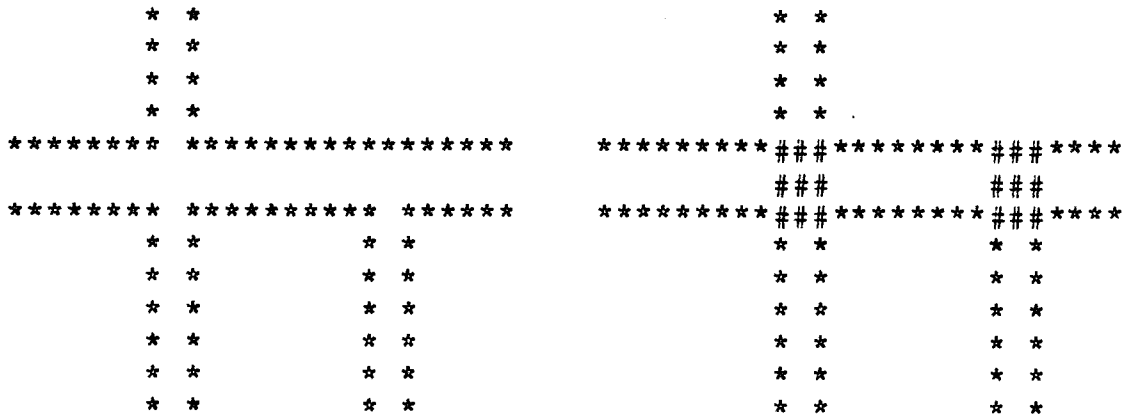
Table 7.1 itemises scanning systems known to the author and which have been evaluated (by the author) for Ordnance Survey large scales applications. Their approach to data handling, after the scanning stage, take on two possible

strategies: some systems offer immediate conversion of the raster output to vector form for validation, feature coding and storage, whilst others preserve raster data throughout (with an option to convert to vector if required). Clearly both strategies have advantages although neither are completely tailored to the OS large scales. The Landranger (1:50000) series or smaller are more appropriate to effective manipulation in raster since they contain both colour and total symbology. The interactive placement of symbols and colour recoding is well-handled by raster work-stations. Large scales mapping presents rather more problems for any raster-based hardware and software. Clearly the scanners can capture the monochrome data successfully (twenty lines per millimetre is appropriate for OS large scale plans) although any of the post hoc reformatting and manipulation is arduous. There are a number of work methods using raster methods open to us:

1. Pass the binary raster image through a conversion to vector process with all its associated problems and then edit/feature-code the vector file on an interactive vector-based graphics system. This has been the approach taken by almost all available systems and is associated with an often inappropriate raster to vector conversion process (see chapter seven).

2. In the raster context, lines drawn on the maps will be several (typically six or seven) pixels wide and hence can be considered as elongated areal units. Feature codes can be assigned to these areas in the raster file interactively. In this way, features can be coded in the raster file by the identification of their start and end locations. This will ensure that the recoding of specific linear map features does not 'ripple' throughout the entire network of interconnected pixels. Conceptually, at least, it will become necessary to detect nodes in the raster line work (where two or more lines join). These must then be temporarily masked as feature terminators to ensure that feature code assignment is restricted to a specific region of pixels. See figure 4.1 below:

Figure 4.1: Map feature delineation by nodal end-points



Within limitations, line weight can assist in the automatic assignment of feature codes. During the set-up stage for the scanning device, threshold parameters need to be set to define the width of lines which the scanner will 'see'. This stage clearly has serious implications for the quality and accuracy of the image emanating from the scanner. Two raster files of the same map image could have very different characteristics if the scanner is calibrated by different operators on each occasion. In practice, human operators can become 'tuned' to the types of threshold settings required for a particular map image (Smith pers comm 1980) which should relieve, to some extent, the problems of the sensitive calibration of the scanner. Weber (1980) and Loodts (1981) have shown how software can categorise line weights in a raster image to provide feature coding assistance.

From an Ordnance Survey viewpoint, pecked lines which comprise approximately twenty per cent of line work in large scale maps (Adams 1979) will cause the biggest problems. With the exception of recent results emanating from the USGS (Smith pers comm 1980), there appears to be difficulties in detecting and connecting pecked lines in a raster image (Scitex Corp. pers comm 1981). The author considers this not to be such a forbidding problem on a parallel computer as outlined in chapter six and, indeed,

the use of push-broom scanners (section 3.3.2) - which can 'see' several scan-lines simultaneously - is also expected to offer robust digital solutions in the future. In essence, this immediate inability to categorise pecks in the image suggests that each peck must be either coded or connected individually; this is possible at the edit stage but clearly too laborious to be a worthwhile proposition.

3. The third approach is likely to be the most interesting academically at this stage and has been the basis for much development in this research project (see chapter six). This involves coding areas (polygons) as clusters of raster picture cells rather than as boundaries (line work). Such an approach will be easy on an interactive work station in raster mode and could have the basis for semi-automated feature coding of raster data on a parallel computer such as the ICL Distributed Array Processor (Parkinson 1978). The resulting data structure will be a true raster representation of the topographic information.

4.2.2 Data manipulation

The lack of any clear specification of user requirements for a topographic data base makes the drawing up of a list of manipulative functions difficult. That such a situation should occur is allied to the need for user education of "what is possible" in a digital environment. Glibly, it is reasonable to expect that some data handling is more appropriate to one data notation or the other. Clearly the algorithms associated with conventional vector data are well enough advanced to facilitate many routine tasks; nevertheless, few attempts by academic geographers (other than the work of Peuquet, 1979a) have been made to adopt a raster solution. To handle numeric data in a grid or array form is perfectly acceptable in computer processing; indeed most high-level programming languages readily cater for this as the most efficient method for storing and processing large batches of numbers.

4.2.2.1 Interface to other data sets

It is the ease of overlay with other data sets which is showing most promise at the present time. Remotely-sensed data from Landsat imagery is a major example although others abound. Benny (1979) has demonstrated the ability to obtain

coastal definition from Landsat imagery. In doing so he outlines several manipulative tasks which can be performed as easily in raster mode as is the case with vector-based notations: geometric transformations of the raster image to specified map projections are possible based on a series of accurately fixed ground control points. This leads to easy overlay with map data for comparison purposes. Any transformation process which involves a translation, change of scale and rotation of an image is easy in raster processing.

Population census data by grid square aggregation can be quickly interfaced with topographic base data for decision making and planning. The American Domestic Information Display System (DIDS) in operational use at the White House currently holds some 500 official statistical data sets organised in some 3000 US counties (Marble and Peuquet 1980). DIDS, which is a raster system, provides data overlay capabilities and retrieval at random as colour maps within six seconds of the query being made to the data base.

Increasingly, the handling of spatial data in grid-based notation is becoming more commonplace in other fields: the Soil Survey of England and Wales are presently investigating the feasibility of storing soil types in raster form (Ragg pers comm 1981).

Elevation information held as raster quantities is seen as having much potential (Adler 1978). Hitherto, the use of Digital Terrain Models (DTM) in the UK has been far less widespread than has been the case in other countries of the world, notably the United States (McEwen et al 1979). This is due, most probably, to the lack of suitable data sets of this nature in Britain with the exception of a DTM of fairly crude resolution developed at the Mapping and Charting Establishment of the Royal Engineers (MCE RE) at Feltham, Surrey. Terrain information in this format is useful for many applications for mapping and planning purposes. Adler (1978) gives a good overview of the organisation and structure of DTM's in land information systems from a theoretical standpoint. For graphics applications, the ability to generate contour data from DTM's is highly desirable - a specific form of raster to vector conversion. Yoeli (1977) gives a comprehensive description of the problems and theories involved. From a more practical standpoint with special reference to planning, DTM's are being extensively used in communications research. The work of GEC-Marconi in Chelmsford - a large electronics group specialising in most forms of communications, defense and industrial electronic systems - is worthy of note. From a communications point of view, the propagation of television and radio waves are highly affected by ground terrain. In order to establish the best site for a TV/radio transmitter

it is necessary to have a reliable intelligence of the surrounding topography in the form of height values over a regular grid. Since no data set to the required fine resolution is readily available in digital form, GEC-Marconi engage themselves in the conversion of graphic contour data to DTM format by manual methods. Thousands of applications of this nature are processed each year for such services as taxi-cab radio control. They consider their manual approach to be highly accurate and since no large scale digital contours to their required standards are available at present, they foresee the continued use of manual conversion for, at least, the next three or four years (Blythe pers comm 1981). Clearly this is just one example of the need for more extensive applications into the mass digitisation of large scales data including the 1:10000 contours in raster form. Such a service should be included as part of a total land information system on a routine basis (Adler 1978) to be generated for repeated and multi-purpose use and not merely as a one-time vehicle for contour generation. The use of photogrammetric methods will assist in this type of work (Petrie 1981), giving further support to a totally interrelated Land Information System as argued by Dale (1977) for many years. The ability to interface information emanating from all sources with the possibility of overlaying all spatial data types is seen as essential in any working system. A raster approach will offer this connection easily

by virtue of the common link of symmetrical picture elements in all the data types.

4.2.2.2 Analytical work

Clearly any type of data overlay is trivial in raster but far more arduous in vector form, requiring detailed cross-referencing of features and carefully planned topology. In a raster image it is the implicit ability to access neighbouring pixels which has the most impact on algorithm design. Pequet (1979a, table 2) itemises some presently available raster algorithms for data manipulation: ironically the majority have their origins in image processing. The list is not exhaustive but provides adequate evidence to suggest that the development of further options, in a raster system, as and when required, will have much promise if carefully planned and explored.

From an OS viewpoint, the calculation of areas and periphery length for polygon data is straightforward in raster; a count of the pixels inside or around the edge of the polygon gives an indication of areas or boundary lengths within certain constraints, which are dependant on the resolution of the pixels and on the orientation of their connection. To perform the same task with vectors requires

more complex arithmetic operations, resulting in slower responses from the computer. The inclusion of area data in OS 1:2500 mapping presently requires a department within OS, which is dedicated to the measuring of polygon areas by semi-automated means. All 1:2500 scale maps now published by the department show a reference and a value (in hectares and acres) for each area measured. These areas comprise such parcels as fields, sections of built-up regions and so on. In the measuring process, the draughtsmen use an Automatic Reading Planimeter by which they trace the perimeter of each area. This is coupled to a counter-unit and a teleprinter which provides the measurement information as a print-out for easy reading and a punched-tape to enable entry to the mainframe for the actual area calculations to be made. To perform this sort of work by manual means is slow and costly. To automate the task further is not without problems: any areal-based calculations with boundary data in vector form tend to be toilsome. The classical example of detecting whether a point described by a single coordinate pair lies within a polygon boundary now has many established vector-based algorithms (Aldred 1972, Baxter 1976), although the solutions are far from simple requiring extensive computing by the central processor. To execute a similar task in raster is simple; the ability to search in all directions from a given point for the polygon boundary makes for an efficient and straightforward solution.

4.2.2.3 Information generalisation

The generalisation of map data for plotting at smaller scales, text placement and so on has been an area in need of development in recent years. Specific problems exist with vector data (Searle et al 1979) which, according to Weber (1980), can be performed automatically far more easily with raster data. Any such generalisation process requires some form of intelligence about the surrounding areas: this is difficult to achieve in vector form but implicitly available in a raster image. City areas, for example, can be generalised to a single point very easily according to Weber, almost by automatic means but certainly interactively with appropriate software. This would be difficult with the OS large scales data but has potential for generalising from 1:2500 to 1:25000 more neatly than with the current vector strategy (see Searle and Waters 1979).

With technology advancing rapidly in this area, it is essential to monitor developments closely. The whole idea of raster storage begs the question 'should we store the topographic data, or representations of that data?' It is possible, for example, to store actual symbols within the data structure: they could be held within the master file containing the raster image or as a cluster of pixels in a pattern library. Intrinsically it would seem odd to hold

representations; it is usual to generate symbols and representations of the data at the output stage. Certainly for a smaller scales data base such an approach seems promising for the short term at least. The storage of graphics symbols within the data will aid the generalisation process although this whole concept implies the need for complex pattern recognition processes at the data capture stage. Undoubtedly, the nature of the information stored within the computer, must reflect the processes to which the data structure will be put.

4.2.2.4 Revision of the data content

The ability to update an existing spatial data file with relative ease is essential for any digital system to remain useful. Theoretically, a raster-based method will provide a solution which is easier and more efficient than amending and replacing a series of vector strings with their associated cross-references to other vectors. Techniques used for the revision of current files in the Ordnance Survey data bank are lengthy and cumbersome. Line work and the position of text often need to be redigitised and then the (laborious) edit cycle begins once more (see 2.5.3) as lines are checked, amended and so on. At worst it becomes necessary to redigitise the whole sheet when the changes necessary cannot

be handled successfully by the piecemeal edit mechanisms. Once the file is brought up-to-date successfully, it is restored in a new location in the data bank; this becomes the child to the parent file which is also preserved for security purposes. Ironically - since the system is based on magnetic tape archiving methods - this involves transferring a whole block of many map sheets across to a new storage volume. Presently, little work has been carried out on attempting to revise the developing low-level link and node files (see 2.7.1). It is anticipated that this will require redigitising as a pre-requisite to the complex restructuring of the new information before transferring it back to the link and node data base. Data in the link and node files are aggregated by 100 metre grid square units, and any revision of data within a grid square will necessitate complete restructuring of that 100m square.

The revision of a raster file need not be such a forbidding problem; the process involves merely replacing a collection of pixels with a new set. Theoretically, this will be a straightforward operation: the updated information is recoded as raster on a scanner and then with appropriate masks it can be overlaid with the existing data. No complex searches or lengthy arithmetic operations are necessary since the whole process is reliant on a series of logical bit comparisons; pixels which are unchanged can remain and new

elements can replace the old. The simplicity is evident since a raster data structure requires no reference to other files as is the case with the link and node strategy. Kummerly and Frey AG, Switzerland, are finding the use of raster methods based around a Scitex Response 250 system to be cost-effective for the revision of the graphics images of their existing small scales mapping (Kummerley and Frey pers comm 1981). These maps are multi-coloured and well suited to the facilities offered on the Scitex interactive work station (see chapter seven). Their uses of the system are purely for map production purposes: they do not store the information for other tasks although such a facility could be a logical extension in the future.

4.2.3 Data presentation

It has been envisaged that map data and spatial information in general will be required in hardcopy form for the foreseeable future (HMSO 1979, section 10.10). Hitherto, hardcopy computer plotters have evolved around the incremental pen or light spot principle. This form of plotting architecture is well advanced and any newly released models merely provide faster plotting rates, easier interfaces to computers and additional colours or plotting options. The nature of these plotting devices has well

suited the vector notation of the data structures in use at the present time. The quality and technical specification of the expensive flat-bed plotters operational in most major mapping agencies - e.g. OS (Searle et al 1979) and USGS (McEwen et al 1979) - are sufficiently well-developed to enable the production of map-drawings suitable for publication. It may well be the case that these types of device will remain as the final copy producers for the foreseeable future, although the rapid development of raster-driven (colour) computer graphics technology in recent years suggests that this alternative strategy has serious potential in future mapping applications. In either case, the methods of data entry and presentation should not dictate the techniques adopted for the organisation of the data structure (see section 2.4). It is clear that robust raster-to-vector and vector-to-raster processes must be developed now to ensure that the best method of data input and output can be taken to suit all needs.

Tests made by the author have suggested it to be inappropriate to plot raster data directly on to vector-based output devices. The examples of raster plots given in chapter seven were produced on a small flat-bed Hewlett Packard vector plotter. This device was used since no raster-based plotting devices were available in the University at that time. Plotting time is unreasonably slow

which arises from the inefficient nature of the pen movement. Furthermore, to attempt to plot raster scan lines on an incremental device is taxing the hardware mechanisms to its operational limits which is bound to reflect in machine failure after a short time.

Theoretically, it is possible to plot raster-based information on a vector-based graphics devices such as those manufactured by Tektronix Corp. and others. The resolution of these devices is high, the images are flicker free and although the contrast is rather poor, the quality of the displayed image is generally good. Since the costs of these devices tend to be moderate, at least for the smaller-sized tubes, it is tempting to look upon them as being sufficiently flexible for the display of both vector and raster data. Investigations have shown this not to be the case. Using a Tektronix 4014 nineteen-inch graphics terminal attached to a dedicated DEC PDP 11/34 minicomputer, tests were made to plot a 1:1250 scale Ordnance Survey plan in raster format. The storage tube can, effectively, be treated as a 4096 x 3123 raster plotting area, although each raster pixel overlaps completely, its *four* neighbouring pixels (Tektronix Inc. 1974). Taking an OS sheet of 434000 set raster pixels and forty metres of vector line work, using a multi-user RT-11 BASIC system, it took two hours to convert the vector data to a raster format and a further 2.3 hours to generate a plot

descriptor file of the results. This is a file which can be interpreted by the graphics device to provide all coordinate information, size of plot, intensity of plotting beam and so on. The elapsed time for plotting was disappointing: it took circa fifty minutes to plot the raster image on the storage tube operating at a speed of 2400 baud, compared with about five minutes to plot the vector file. The quality of the resulting raster image was encouraging but, nonetheless, processing and plotting time is unsatisfactory. Further trials at plotting only selected pixels reduced the time lapses by a factor of four or more without serious loss of image quality. The slow processing times quoted for plot generation are a function of the inefficiency of multi-user BASIC under the DEC RT-11 executive and to a certain extent of program code which has not been optimised. Yet, despite this, it can be concluded from the exercise that the use of vector storage tubes is not appropriate for the rapid display of raster images.

Clearly the efficient display of raster files for edit purposes or other tasks requires appropriate graphics hardware. Hardcopy raster printer/plotters and monochrome or colour raster display screens are becoming more popular as computer graphics systems gain importance and reach maturity. Raster-driven graphics displays are now well-developed, relatively inexpensive and well understood, since basically

the technology which is employed in the actual display device is similar to that used in domestic television sets. They are potentially more flexible than the vector storage-tube devices since the problem of displaying vector coordinates and continuously refreshing the display has recently been overcome with the availability of suitable, inexpensive micro-processors. The result has been a dramatic fall in the price of these devices over the last two or three years. Many raster-driven refresh graphics displays have appeared on the market at a price well below that of comparable sized storage tubes although their resolution is presently inappropriate for most mapping applications.

The production of hardcopy displays for raster images are probably not as well advanced and comparably inexpensive as the display screens; they involve far more by way of moving parts. Nevertheless, viable devices are available and intense development abounds. Currently available devices offer plotting capabilities of typically 1000 lines per minute with high resolution (100 or 200 points per inch). They operate on an electrostatic principle and are capable of holding chart widths between 8 and 42 inches, depending on the model chosen: length is limitless since the paper is held on a continuous roll. These types of device, manufactured by Versatec Electronics Ltd. are well suited to edit or development work. Experiments carried out for cartographic

purposes are encouraging (see chapter seven). Serious alternatives are being offered by laser plotting techniques for applications requiring higher quality raster images comparable with those being produced from the extremely accurate vector plotters. The laser exposes the raster image on to high-contrast, medium speed, lith film at high speed. Detailed consideration of raster dedicated hardware is given in chapter seven. It can be concluded here that developments in raster entry and display systems abound and are beginning to provide serious alternatives to the vector systems which have monopolised computer graphics for a decade or more.

4.2.4 Data storage

It has been shown in chapters two and three how data structures can be developed to suit both vector and raster-based spatial data. In most geographic data structures quite complex vector representations have been developed. The Canadian Geographic Information System (Switzer 1975) was probably one of the first and even today seems to have hardly outdated itself. Others such as the US Census DIME files (Cook and Maxfield 1967), POLYVRT (Peucker and Chrisman 1975), GEOGRAF (Peucker and Chrisman 1975) and others are all extensions of these well established line segment or polygonal ideas. More recently Shapiro et al

(1980) has attempted to compare and contrast these systems. It is anticipated that all spatial data stores on a national level will be voluminous (see Adams et al 1981 for the Ordnance Survey example): initial experiments with raster-based equivalents (see chapter five) suggest that volumes will need to increase two or three fold. It may well be the case that to store topographic data in raster form is too costly in terms of backing store requirements. This will surely not be the case if the innovations promised by the computing industry have any substance. One need merely glance through the computing press each week to read of the planned high-density archival storage devices to be released in the next five or ten years. Computer bubble memory, video disks, holographic storage and so on are all being developed. Some non-magnetic archival systems under investigation are capable of holding 10^{10} to 10^{12} bits per square inch and with low error rates, of the order 10^{-9} or better.

Clearly the storage of raster information is not going to cause too many problems for the computer industry: the simplicity of the data structures involved (see 3.2) are an added stimulus to continually monitor hardware developments. Nonetheless, the ability to 'flip' between raster and vector or vice versa on a routine basis is seen as essential. The two methods are not seen as inviolate: they must be used in conjunction for any system to remain flexible. The

mechanisms for such an approach (Boyle 1979, Peuquet 1981, Harris et al 1981) have been considered in a somewhat piecemeal fashion in the past although it is difficult to improve upon this until a thorough knowledge of data characteristics and usage has been assessed. This, to a certain extent, must remain with the mapping agencies and associated research workers who, in collaboration with map users, must tailor a system to suit the majority needs.

4.3 SUMMARY AND CONCLUSIONS

This chapter has attempted to distinguish between raster and vector methodology in digital cartography. It can be concluded that although vector methods are fairly well established, there tends to be much scope for the development of raster techniques to aid the existing systems. Raster-based data handling of topographic information offers a serious alternative to the presently accepted approaches for the capture and management of spatial data. With continued exploration and applied research there is every reason to expect that grid-based approaches will overtake existing methods with reduced costs and higher throughput rates by the 1990's. Clearly, any statement of development must remain speculative but it can be argued that raster technology is merely trailing its vector counterparts through

lack of serious investigation. With the advent of working raster systems and computer peripherals (see chapter seven), existing raster know-how and ideas on future developments (see chapter five) and the introduction of extremely powerful array processing techniques and hardware (see chapter six), it remains only a matter of time before these methods play a major part in any working information systems on a routine basis.

Thompson (1978 p10) has assessed comparative costs for a raster and vector production flowline for the Ordnance Survey large scales digital mapping system. The unspecified raster system is anticipated as marginally more expensive than a Laser scan Fastrak flowline. In 1978 this may well have been the case: certainly raster technology was far less advanced than the vector methods at that time. There is every reason to expect a reversal of roles in the medium term future. It will be easy to attempt to replace existing systems completely with these new methods. Such an approach would be wrong; both have important roles. It is seen as crucial that robust interfaces should be established between the two methods. Dual-efforts by both systems will offer many benefits when fluent conversions between both raster and vector notations are possible.

Hitherto, no attempts have been made to handle OS data in raster formats. Unequivocally it is necessary to investigate the suitability of these present and future techniques with respect to the characteristics and requirements of OS workloads. In the next three chapters, reports are made on the nature of these tests carried-out, to discover more about raster processing and its viability to specific mapping problems.

5 THE MANIPULATION OF RASTER DATA ON A SERIAL COMPUTER

5.1 INTRODUCTION

The use of electronic digital computers in the environmental sciences has been widespread for almost two decades. The applications have been twofold: they have the ability to process vast and complex calculations at high speeds and they are able routinely to store and access large quantities of information. Their computational capabilities have enabled analyses and calculations previously taking several man-days to be performed in a few hours or less. Davis et al (1972) provides ample evidence of the effects of computational power in the readjustment of the 1970 scientific triangulation network of Great Britain. Conversely, the use of digital techniques for the storage and maintenance of spatial data to develop fully comprehensive land information systems (LIS) requires different strategies. The use of computer installations to host data base management systems has tended to be less concerned with computational power and more tuned towards efficient data structure and transfer. This has been reflected in the nature of the traditional programming languages: FORTRAN and, to a lesser extent, ALGOL are extensively used for scientific

work whereas most administrative and data base applications are written in COBOL. The development of modern block-structured languages such as PASCAL and APL suggests that computational power and efficient data structure are becoming equally important in more recent applications.

Hitherto, it has been the speed of the transfer of data between main memory and backing store (magnetic disk or tape) which has been of primary concern in the handling of geographic data (Marble et al 1978). Data base systems are less concerned with high computational rates since very little arithmetic is required with the exception, perhaps, of variable comparisons or the amalgamation of data to produce new variables. This heavy reliance on data transfer is a disadvantage. Modern computers can perform arithmetic operations very quickly; their performance is, more or less, constrained by the speed of electronic signals in the circuitry of the central processor. The speed of a serial computer can be stated in many ways: two of these are "number of multiplications per second" and "transfer time". Modern machines can perform up to 500,000 multiplications per second. The transfer time is the time taken to transfer a number from a storage location in main memory to the arithmetic unit (the area of the central processing unit where all arithmetic and logical operations are performed). Three microseconds could be a typical figure for such a

transfer although faster machines have a transfer time of less than 500 nanoseconds.

The time required to access data elements at specific locations in the backing store is dependant on moving parts and is much slower and less reliable. To extract information from a magnetic tape requires a severe time lapse whilst the drive spools the tape to the designated position; the time lapse is dependant on the nature of the data blocking and on the characteristics of the hardware, although ten to fifteen minutes is not atypical to read a full standard 1600 bpi 9 track 2400 feet tape on a IBM 3420 tape drive. The same drive is able to rewind the tape in twenty or thirty seconds but it is unable to read at these rates. The transfer rates associated with magnetic disks are faster than with tapes. Each surface has a number of concentric tracks on which information is stored at a unique address. The time lapse to retrieve a specific data element is a function of the rotational speed of the disk and the linear movement of the heads to reach the desired track. Whilst access times are measured in milli-seconds rather than in micro- or nanoseconds associated with main core, the larger capacities and lower costs associated with backing store makes them more attractive for mass storage and high data volumes. The adoption of modern winchester technology (see Systems International, Oct 1980 pp51-52) in which the read/write

heads and disk units are combined is enabling faster retrieval and higher data densities to be possible. ICL have been leaders in the development of "intelligent" backing-store controllers. The Content Addressable File Store (ICL CAFS) (see Maller 1979) has the ability to filter the information before it is passed to the central processor, thereby reducing the chance of transferring inappropriate data and speeding-up retrieval times from a complex data structure. In general, however, there are no indications that the pattern of the last 20 years will not be continued: thus, further development and increased performance of such computer storage technology can be expected to alleviate (but certainly not obviate) the problems associated with large geographic data sets in the next decade.

Perusal of the appropriate literature suggests that the introduction of raster techniques for the handling of topographic data has generated a dichotomy in existing skills and resources. Grid-based equivalents to the complex vector data structures are more voluminous and also tend to require far more by way of computational needs (in particular, data transfers and arithmetic operations). For example, the process of converting vector-to-raster-to-vector is highly central processor dependant (Harris et al 1981, Peuquet 1981). These factors suggest the need for carefully controlled algorithm design and data volume.

Detailed use of raster data on such serial machines as described above has been carried out by the author. A number of specific experiments have been undertaken which have three distinct aspects: their design, purpose and results are set out in the following three chapters.

All three aspects of the experiments have involved attempts to discover the potential benefits of a raster methodology for the OS large scales topographic data. Until hitherto, there has been no formal investigation into the uses and applicability of raster methods for Ordnance Survey data (see section 4.4). This chapter describes the feasibility of handling such raster data on conventional serial processors. The work considers the digital conversion processes, data compaction techniques and so on.

This series of experiments concludes that raster-based data storage and manipulation of Ordnance Survey data is relatively costly; the data volumes will increase over those associated with vector methods and far more manipulative operations are involved which are cpu-intensive. There is much evidence to suggest that raster-based cartography does not fit neatly into the architecture of conventional serial computers (see section 5.2). The necessity to perform identical operations on each pixel in turn is not ideally suited to sequential processing. A series of identical

central processors working simultaneously is expected to increase throughput rates one hundred fold or more; chapter six describes the design and nature of some raster-based manipulation using modern array-processing techniques on an ICL Distributed Array Processor.

However good the data manipulation capabilities of an array processor, appropriate digitising capabilities are clearly essential. The third phase of experiments has involved appraising the suitability of commercially available raster-scan systems for digitising, then manipulating and plotting OS 1:1250 data. The study and its results are outlined in chapter seven.

5.2 A SERIAL-TYPE COMPUTING ENVIRONMENT

Comprehensive descriptions of the workings of modern computer systems can be found in appropriate manuals; Nelder (1975) and Baxter (1976) provide simple descriptions of the various parts of a computer system. However, a description of the new generation of parallel-type processors is given in section 6.2 and, for comparison purposes, an outline of the conventional serial-type central processor will be given here. The heart of a traditional, sequential computing system is the central processor; this unit houses three

constituent parts - in addition to the main memory (or core store), there is a "control area" where all programs are stored and an "arithmetic unit" where all the arithmetic is performed. The architecture is serial or sequential in nature because the arithmetic unit can only perform one calculation at a time. For instance, when adding two numbers together, the control locates the address of the two numbers in the core store, transfers them to the arithmetic unit for adding and returns the result to the desired location. Repeated addition of an array of numbers as may be necessary in raster data will involve a continual cycling through this operation for each number in turn. Electric pulses pass through computer circuits at the speed of light (298,000 km per second) and the time taken to run a particular program on a computer is a measure of the distance travelled by electrical currents inside the central processor. From being based on thermionic valves to printed circuits to integrated circuits, computers have become physically smaller, hence distances travelled are less and computational speeds have risen. Given present miniaturisation, it seems highly likely that the performance of future raster-based systems will depend primarily upon the volume of data involved. Since the internal mechanisms of computers cannot be faster than the speed of light, any upgrade in system performance so far as raster-based cartography is concerned must either involve a reduction in the volume of the data being processed or the

use of a series of central processors working simultaneously (see chapter six).

All existing raster research, with the exception of ETL, Fort Belvoir (Babcock 1978), has adopted serial-based algorithms with attempts to keep data volumes in perspective with the use of complex data compaction and storage methods. Most commercially available raster systems are based on the popular mini-computers, typically Digital Equipment Corp. (e.g. Kongsberg Kartoscan or those systems emanating from Broomall Industries Inc.) or Hewlett Packard (e.g. Scitex Response 250). In reality, the power of these mini-computers is now as great as that of many mainframe machines. Indeed, the distinction between mini-computers and the traditional mainframe seems to have become virtually non-existent in recent years; comparatively cheap stand-alone machines today have as much computational power as vastly expensive systems of the early seventies. Modern distinctions tend to be made between machines requiring an atmospherically controlled environment with experienced operators and the relatively robust dimensionally-smaller machines used as part of more specialised devices: a raster scanning system is an example.

The serial-based computing system used for these experiments was the Northumbrian Universities Multiple Access Computer (NUMAC), the only machine available to the author

with adequate peripherals such as ample disk space and tape drives. The computing system is based around an IBM 370/168 processor with eight megabytes of virtual memory and a 32 bit word length. The operating system is the Michigan Terminal System (MTS), a versatile executive supporting up to 150 users in a multi-user environment without serious loss in system performance. All software developed for experimental purposes was written in FORTRAN IV optimised where necessary on a H-level extended IBM compiler; some assembler coding was used where necessary.

5.3 THE DEVELOPMENT OF A VECTOR-TO-RASTER CONVERSION SYSTEM

Several research workers (Peuquet 1981, Harris et al 1981, Weber 1980, Boyle 1979, Babcock 1978) have already devoted much time and resources to the investigation of adequate conversion techniques to move fluently between the two data notations. However, all existing systems are piecemeal; they have developed generally for special classes of data only (e.g. for thematic polygon mapping, contour mapping, etc.). Certainly, no reasonably integrated solution is known to exist at present for the complex OS formats. Thus, experiments to convert existing OS vector digital data into a raster mode was seen as essential for the following reasons:

- There were no systems available to transform the existing OS vector-based data structure into a grid-based representation.
- It was considered desirable to decant the existing OS feature coding scheme in a raster file; no scanning system generally available can provide this facility at the present time.
- Such an exercise would provide a control data set with which to compare results obtained from scanners.
- Such an exercise would provide provisional data to facilitate investigations and design strategies with parallel algorithms.

5.3.1 Design aims

These are outlined below.

1. To ensure that any conversion software is as general in nature as possible. Specifically, input data formats should not be restrictive - pixel resolution should be parameterised for example.

2. To generate 'complete' raster (i.e. uncompact) data as output from the software.
3. To compact these as a secondary stage when necessary. This strategy facilitates investigations into data characteristics, serial processing times and gives a clearer appraisal of the problems and benefits of adopting a raster strategy for OS-type data.

To handle raster-based information in its uncompact form generally is resource-intensive. Each 1:1250 scale OS graphic measures 400mm x 400mm; the minimum line weight used is 0.18mm. Hence, to represent each map sheet with a pixel resolution increment of 0.1mm (c. half the minimum line width) requires a raster buffer of 4000 x 4000 pixels. The choice of pixel resolution must compromise between graphic quality and available computer storage; the choice of half the minimum line-width was an initial compromise between these two factors. The inevitable 'stepping' effect of line-work is a serious disadvantage although no perception experiments are known which appraise the smallest step-size discernable to the eye. Initial assessments of the storage requirements for 0.1mm pixels are large, as shown below.

1. to store as a bit map
(1 bit per pixel) = 2 megabytes
2. to store as a byte map
(LOGICAL*1: 8 bits per pixel) = 16 megabytes
3. to store as a halfword integer map
(INTEGER*2: 16 bits per pixel) = 32 megabytes

To preserve the existing feature coding scheme in the conversion process requires at least one byte per pixel; each element can then contain either zero for no data or 1-127 feature types (there are currently 109 attribute types in the OS scheme). In IBM FORTRAN, the smallest unit of store with which it is possible to perform arithmetic operations is the halfword (16 bits on the IBM 370).

Early tests suggested it to be most convenient for users to organise the data in units of ground metres instead of plot coordinates at output scale. If each pixel increment represents 0.1 metres at ground scale (0.08mm at 1:1250) then, to preserve the 'half minimum line weight' criterion for pixel resolution requires the initial raster buffer to be 5000 x 5000 pixels. Originally, tests were based on each pixel comprising a half-word integer to facilitate straight forward FORTRAN programming initially. The storage resources for this raster buffer were fifty megabytes, which is in

excess of 12000 MTS pages (the basic unit of disk storage under the MTS executive is 1 page = 4 Kbytes). Clearly, even the very large virtual memory of the IBM 370 cannot process this amount of data in one pass and hence it is necessary to partition the buffer into strips or chunks.

Franklin (1979) gives a comprehensive summary of the methods of partitioning large raster images on a computer system. Most of the techniques which he has suggested involve an iteration of various algorithms over a series of consecutive strips, only one of which can be held in core at a time. In the vector-to-raster conversion process, for example, the simplest approach is to examine sequentially, the vector data file and convert any of the vectors which lie in the current raster strip to a pixel notation. This procedure is repeated iteratively for each strip. Logical extensions of this basic approach to speed-up the processes are also suggested by Franklin (1979). Examples include re-writing unused pieces of vector lines to a secondary file for subsequent iterations, or pre-sorting the vectors in the vector data file prior to the conversion process and sub-dividing it by pre-splitting the vectors into a series of files which correspond to the number of logical strips in the complete raster buffer. Evaluations of Franklin's ideas were made by the author with OS data in a raster notation (see section 5.3.6). Such trials are inevitably constrained by

local system facilities: it is advisable to restrict the amount of main memory to around a half megabyte on the NUMAC IBM 370/168 under MTS to avoid the degrading effects of page thrashing in a multi-user environment (see Visvalingam 1977), though it is actually possible for users to obtain up to one megabyte of virtual memory under MTS for large jobs. Early tests were carried out in partitioning the 5000 x 5000 raster image into 50 strips (numbered 0-49) of 5000 x 100 dimension. This represents 50 one megabyte images, each based on two-byte pixels: this storage was subsequently reduced by a factor of two when software was written which could handle pixels of one byte each.

5.3.2 Theory

The theory behind the vector to raster conversion is manifestly straight forward; it is the volume of data which produces computational problems. Under Ordnance Survey conventions, each vector comprising the map line work is described by a series of x,y point locations. Appendix 5.1 outlines the mathematical derivation of the equation of each straight line knowing two points on the line. Any conversion of the vector-segments to a raster image requires repeated solution of this derived equation in increments which are dependant on the desired pixel resolution.

5.3.3 Some data transfer considerations

As a result of the study of OS data characteristics (Adams 1979), 478 Ordnance Survey digital map sheets were available in the customer export format (DMC). This data format has been developed to ease the problems of data transfer between the Ordnance Survey ICL 1900 computer site and a wide range of user's installations. Each logical record comprises eight FORTRAN readable characters; it is unreasonably inefficient for any serious development work. In order to make the data input and output (I/O) rates as efficient as possible for an already lengthy conversion algorithm, the DMC characters were reformatted into National Grid (NG) coordinates which could be stored as blocks of data in binary form: this was expected to increase I/O speeds by fifty fold or more.

A special purpose utility program ('FROMDMC') has been written in FORTRAN as a front-end to the vector to raster software. Its intention is to take the standard OS DMC records as input and reorganise them into binary blocks of NG coordinates with their associated OS feature codes. A listing of the source FORTRAN for this utility can be found in Annex 5a. The DMC export format provides a precision of 0.001 of a basic grid square for x,y coordinates: this represents a positional accuracy of 0.05 metres on the ground for the 1:1250 data. Each NG coordinate comprises eight

significant figures with two decimal places; double precision in FORTRAN (REAL*8) was used for storage on the IBM 370. In the large scales data the feature code list never exceeds 109 types and hence half-word integers (INTEGER*2) were adopted for their storage. The utility program groups together one thousand NG coordinates into blocks for efficient use of backing store and I/O transfers. This provides a temporary, intermediate data file for entry into the vector-to-raster software which is formatted as:

```
X(1000), Y(1000), ICODE(1000)
```

where X,Y are REAL*8 NG coordinates;

ICODE is INTEGER*2 attribute code.

A simple technique was used to describe the start and end of each feature within the blocking structure; the first occurrence of ICODE (the feature code) is made negative for the first x,y point. In this way, it is trivial to detect point symbols too.

The results obtained from 'FROMDMC' were encouraging both in terms of a reduction in I/O speeds and a crude form of data compression. The performance of the technique is shown

in Table 5.1 for a Durham 1:1250 sheet (NZ 2642NE) where a seventy-fold increase in I/O performance can be seen.

Table 5.1: Performance of the compression stage for a 1:1250 scale vector file (NZ 2642NE)

Before compaction: DMC character format

Number of records	31,115	
Number of x,y points	14,789	
MTS storage requirements	181	pages
CPU time to read total file	21.796	seconds
Number of read operations	31,115	

After compaction: blocked binary

Number of records	15	
Number of x,y points	14,789	
MTS storage requirements	96	pages
CPU time to read total file	0.309	seconds
Number of read operations	15	

5.3.4 Development of the vector-to-raster conversion algorithm

The initial test was designed to be simple and consist of:

- The raster buffer allocated two bytes per pixel: this required fifty megabytes of backing store which was made available on an IBM 3330 dual-density disk pack with a total capacity of three hundred megabytes.
- The input vector data were read in their entirety for each raster strip, although by far the biggest proportion of the data for each partition will be irrelevant (see Franklin (1979) and section 5.3.6).
- The input data, although in binary form, were not sorted in any way and remained in the same sequence as supplied by the Ordnance Survey.

A listing of the final source FORTRAN code used for the conversion software can be found at Annex 5b. The algorithm reads each x,y point and associated attribute code individually from the input buffer as outlined in section 5.3.3. If the feature detected is a point symbol then the appropriate address in the raster buffer is set with the attribute code for that feature (see lines 80-88 in annex

5b). Linear entities require a more involved approach. The notation adopted is important: the first x,y point is called X and Y with corresponding attribute code L . The next point in the chain (to describe the first vector segment) is read in to give X_1, Y_1, L_1 . The derivation of the equation of a straight line knowing these two points is given in Appendix 5.1. The conversion of the vector segment into a raster mode involves repeatedly solving the equation of the straight line for each increment of the raster buffer. There were four immediate problems in the FORTRAN coding for this algorithm:

1. The repeated solution of the equation must be transformed into a cyclic loop in serial code - the 'DO-loop' in ANSI (1966) FORTRAN. Standard FORTRAN stipulates that the delimiters of a loop must ascend from a lower to a higher number. Thus the values of X, Y, X_1 and Y_1 must be pre-sorted into $X_{MAX}, Y_{MAX}, X_{MIN}$ and Y_{MIN} (see lines 92-100 in annex 5b).
2. Since a DO-loop in FORTRAN must have integer controlling variables, then to handle a 0.1 increment requires multiplying the X and Y values by ten to develop this controlling variable and subsequently dividing by ten once inside the loop.
3. To solve repeatedly the equation of the line for y given

each value of x alone is inappropriate: some pixel assignments are missed out completely for lines with orientations very near to vertical. A robust algorithm must perform the solution in two passes: it solves the equation for y given x first (see lines 127-155 in annex 5b), then performs a vector re-sort and solves the same equation for x given the y values (see lines 157-187 in annex 5b). The two special cases of perfectly vertical lines (running N-S) and horizontal lines (running E-W) must be trapped (see lines 103-120 in annex 5b) and handled separately to ensure no attempts are made to divide by zero in the equation (see appendix 5.1).

4. By definition, each line between two vertices will have its limits described by the end points X_{MAX} , Y_{MAX} , X_{MIN} and Y_{MIN} . When solving the equation for x given y we can be sure that the values of y will never exceed these limits although the computed values of x can exceed them. (A similar situation arises for the y values when solving the equation knowing the values of x). To ensure that these limits are never exceeded, a tolerance threshold has to be defined which has been found empirically to be half the pixel resolution. If the values of y and x at each end of the line are computed to exceed the end points by greater than a half pixel, they are rejected (see lines 135-145, 177-187 in annex 5b).

5.3.5 Development of associated utility programs

At the time at which the experiments were carried out, the Durham University computing service had no means of displaying raster images as rasters (c.f. raster images simply shown as pen up/pen down movements on a vector plotter). For this reason, it was difficult to appraise the characteristics and quality of the resultant vector data and a number of associated utility programs were developed to aid analyses of the results generated: the FORTRAN source for these utilities can be referred to in Annex 5c.

1. LPDRAW reads in the raster image and reduces the resolution suitable for crude plots on a line printer. It prints out a matrix of 100 x 100 elements: each element represents a 50 x 50 cluster of the raster pixels in the original 5000 x 5000 buffer. The program was designed to search for and display only those feature types specified as input parameters. An example of the output generated can be seen in figure 5.2a in which the centre-lines of roads for sheet NZ 2642NE are shown; for comparison, the equivalent vector data is in figure 5.2b.
2. RASTER.SCAN prints out the buffer address of each non-zero pixel.

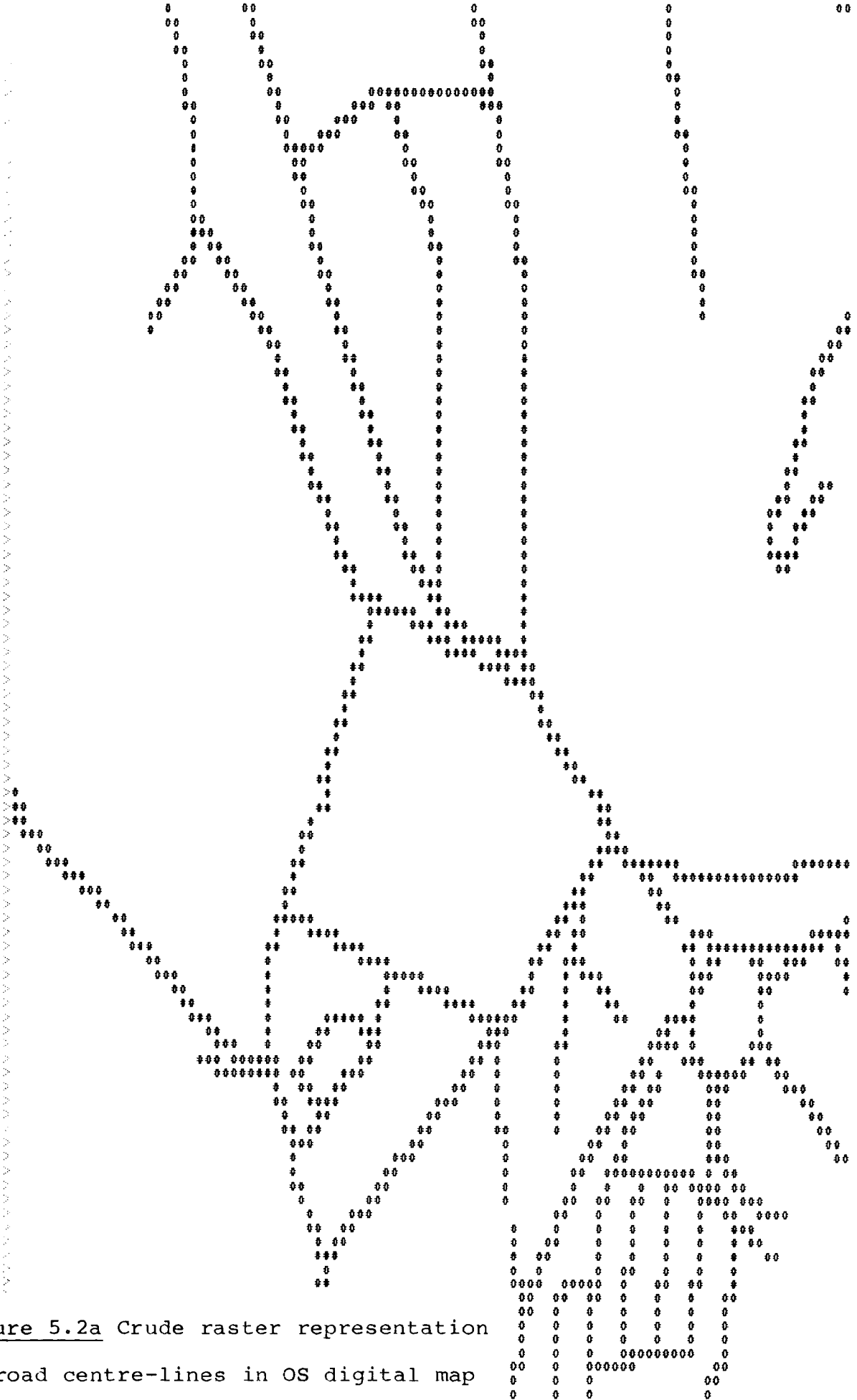
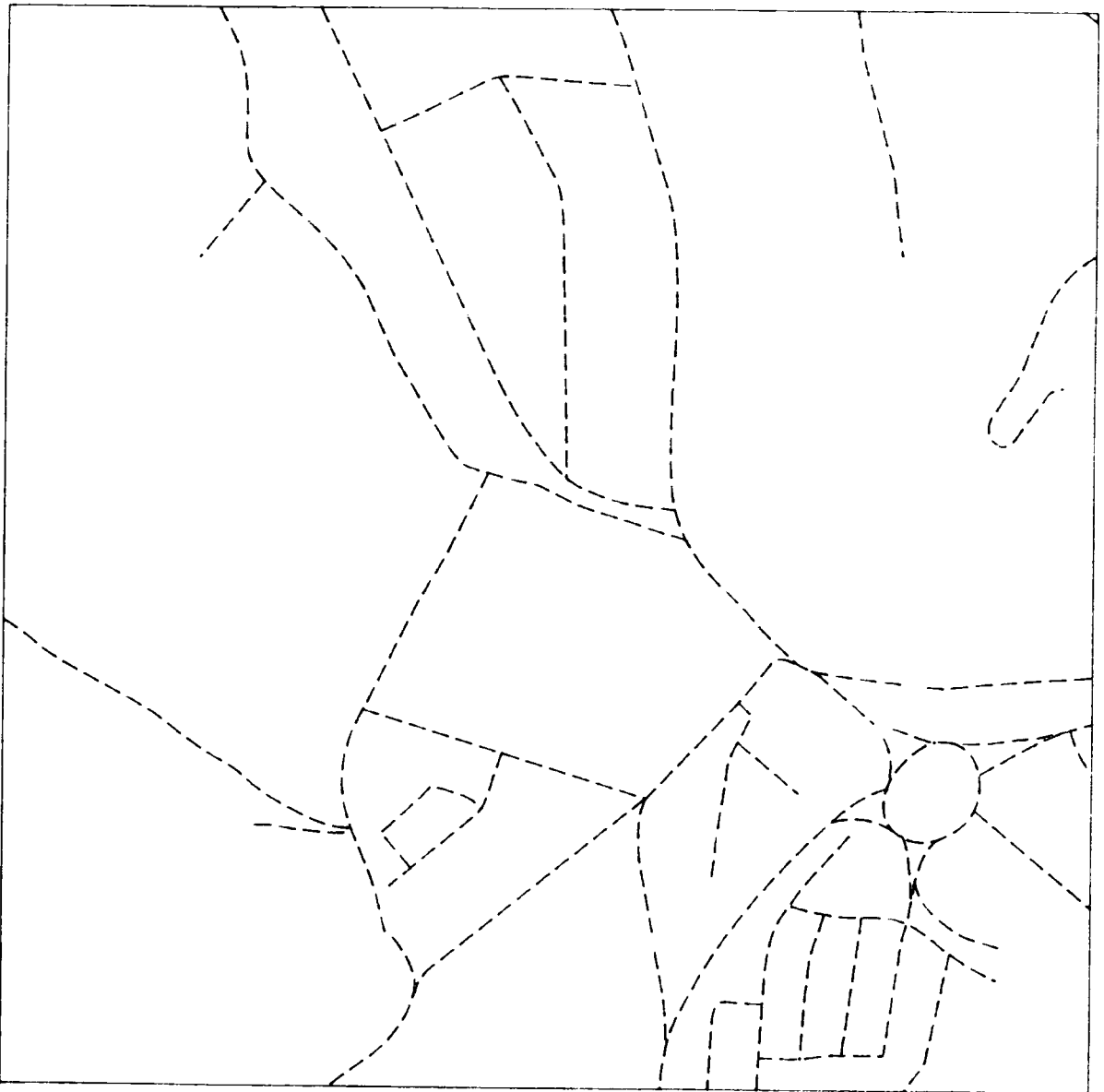


Figure 5.2a Crude raster representation of road centre-lines in OS digital map produced by utility program LPDRAW on a lineprinter. Crown Copyright Reserved



542800.0

07000000

Figure 5.2b Equivalent vector plot of
road centre-lines shown in fig. 5.2a.

Crown Copyright Reserved.

3. RASTER% performs some elementary analyses on the raster buffer; it reports the percentage sparsity of the image and breaks down feature codes into pixel counts. An example of the output from RASTER% is shown in Table 5.3

Table 5.3: Break-down of 5000 x 5000 raster image of NZ 2642NE by utility RASTER%

<u>Feature description</u>	<u>Total line length</u> (mm)	<u>Number of x,y points</u>	<u>Pixel count</u>	<u>Percent set pixels</u>	<u>Percent of total buffer</u>
Empty pixels	---	---	24458155	---	97.833
Public buildings	816.6	228	11445	2.112	0.046
Minor buildings	1176.5	678	15839	2.923	0.063
Other buildings	5665.1	1410	74589	13.766	0.298
Open buildings	216.6	133	3057	0.564	0.012
Archway symbols	59.6	20	716	0.132	0.003
Railway lines	1639.3	134	28372	5.236	0.113
General boundary	347.9	21	6076	1.121	0.024
Railway switch	1.0	2	9	0.002	0.000
Road carriageway	4709.2	1852	63120	11.649	0.252
Road centre line	3560.3	794	47486	8.764	0.190
Footpath	71.8	55	963	0.178	0.004
Control point	---	1	1	0.000	0.000
Bench mark	---	9	9	0.002	0.000
Spot height	---	12	12	0.002	0.000
Road fence/wall	5425.2	1504	71365	13.171	0.285
Genrl. fence/wall	10127.3	3661	140945	26.012	0.564
Road casing	90.4	55	1251	0.231	0.005
Banks/baulks	3720.8	2001	50907	9.395	0.204
Subway	78.5	23	795	0.147	0.003
Vegetation limits	160.6	65	2115	0.390	0.008
GPO call box	---	2	2	0.000	0.000
Step treads	1038.4	1200	13684	2.525	0.055
Point feature	---	5	5	0.001	0.000
Stream	17.0	4	52	0.010	0.000
Ward boundary	493.6	99	3996	0.737	0.016
Pavement casing	376.3	417	5027	0.928	0.020

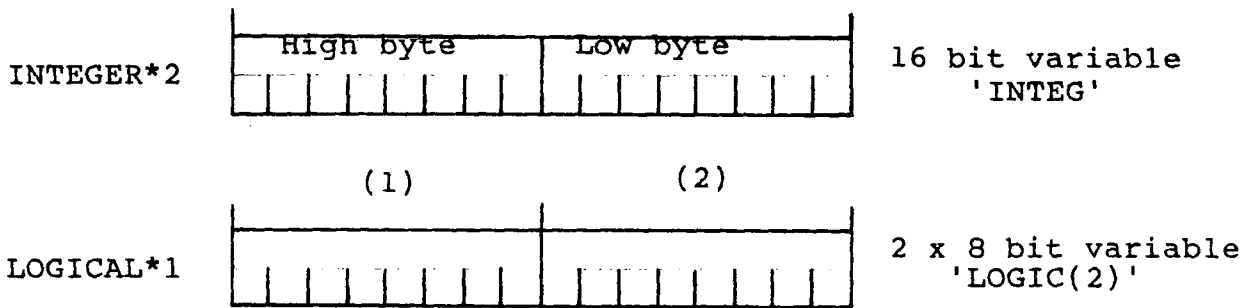
5.3.6 Software tuning

Initial tests were attempted using a file of road centre lines only. This facilitated a rapid turnaround of results and easy validation during the development stages. Once the code was working successfully, a full Ordnance Survey 1:1250 map sheet (NZ 2642NE) was converted to raster form. Out of the twenty-five million pixels in the raster buffer, only 541,845 were set (i.e. had been assigned feature codes). These figures are, of course, highly dependant on pixel resolution, map sheet complexity and so on although they suggest that OS large scales data in raster format are of the order of 97.8 per cent sparse.

The actual conversion process was carried out in fifty strips of 5000 x 100 pixels: the central processor time on the IBM 370 to process each strip was 10.5 seconds. The total time taken to rasterise the map sheet in this crude way was 525 cpu seconds. It is the partitioning of the buffer which has a fundamental effect on the process times; if a hypothetical fifty megabytes of main memory could be acquired then the rasterising would take little over eleven seconds. A reduction in the number of partitions will increase the process times dramatically. Hence, in converting the raster buffer to allocate one byte per pixel, it is possible to increase the size of the strips by a factor of two and halve

the overall storage requirements. In IBM FORTRAN, only logical data (.TRUE. and .FALSE.) can be manipulated within one byte units (LOGICAL*1); it is necessary to modify the code extensively to store numerical data as single byte elements. Figure 5.4 shows the technique adopted to store an integer value in a one-byte (LOGICAL*1) variable.

Figure 5.4 The use of one-byte variables for numerical data



Providing that the value of 'INTEG' in figure 5.4 does not exceed 127 then it will be stored within the least significant eight bits (the low byte). Using the FORTRAN statement 'EQUIVALENCE' it is possible to overlay the two-element LOGICAL*1 array 'LOGIC' with the two-byte INTEGER*2 scalar 'INTEG' thereby storing the contents of the low byte of 'INTEG' in the second element of 'LOGIC'. In this way it is possible to transfer to backing store the second element of 'LOGIC' and still preserve all the information in the raster buffer. A simple program to

transform a two-byte element raster image into one byte per element is shown below:

```
INTEGER*2 IRAST(5000,100), INTEG
LOGICAL*1 LRAST(5000,100), LOGIC(2)
EQUIVALENCE (INTEG, LOGIC(1))
1 READ (1, END=3) IRAST
DO 2 J=1,100
DO 2 I=1,5000
INTEG = IRAST(I,J)
LRAST(I,J) = LOGIC(2)
2 CONTINUE
WRITE (2) LRAST
GO TO 1
3 STOP
END
```

This will reduce the storage requirements by a factor of two although it will be necessary to expand the pixels back into two-byte integers for any arithmetic manipulation. Alternatively, boolean operations such as searching within spatial neighbourhoods for pre-defined patterns or straight line features (e.g. building walls or grid lines) will not require unpacking of the compacted data. The unpacking stage is, essentially, a reversal of the above process. In FORTRAN the left-most bit of an integer variable is the sign bit which stipulates whether the integer value is greater than or less than zero. This is not the case with LOGICAL*1 variables: to handle the storage of negative integers in a

single byte requires a test to be made at the unpacking stage. If the stored number is negative then an untested conversion back would produce a number greater than 127 since the left-most bit would be included. In IBM representations, a negative number has the high-byte set to hexadecimal FF (binary 11111111) and this must be assigned manually where necessary at the conversion stage. A simple program to expand LOGICAL*1 pixels back into INTEGER*2 is shown below.

```
INTEGER*2 IRAST(5000,100), INTEG
LOGICAL*1 LRAST(5000,100), LOGIC(2), FF/ZFF/
EQUIVALENCE (INTEG, LOGIC(1))
1 READ (2, END=3) LRAST
DO 2 J=1,100
DO 2 I=1,5000
INTEG = 0
LOGIC(2) = LRAST(I,J)
IF (INTEG .GT. 127) LOGIC(1) = FF
IRAST(I,J) = INTEG
2 CONTINUE
GO TO 1
3 STOP
END
```

Empirical evidence has shown that the use of this technique enables the compaction and storage of any number within the range -999 to +127 on the IBM 370/168, although theoretically this range should be restricted to -127 to +127. Its use has proved to be highly flexible in a number of raster-based serial algorithms developed by the author

(see later). Evidence has suggested it to be more appropriate to perform the conversion within the main program rather than including a subroutine for the purpose; this is because of the large number of conversions (and hence subroutine calls) necessary. Tests have shown that for each 5000 x 100 strip, 9.4 additional cpu seconds are included in the processing times for subroutine calls against 3.3 seconds for performing the compaction in the main program.

Overall, processing times proved to be slightly improved over the two-byte pixel method; the added processing times for the data compaction are compensated by much faster I/O rates since data transfer between main memory and backing store is halved. Hence, to rasterise sheet NZ 2642NE with the compaction method took on average of 10.1 seconds per strip giving a total cpu time of 505 seconds for the whole sheet (a three per cent increase in overall speed).

With the introduction of a LOGICAL*1 raster buffer, the storage requirements of each 5000 x 100 strip are reduced to a half-megabyte. It is possible, therefore, to increase the strip size to 5000 x 200 pixels and hence reduce the number of partitions for this raster image to twenty five. The virtual memory of the IBM 370 under the MTS executive seems to cope with a one megabyte array successfully using optimised software; processing rates have proven to be highly

favourable. The cpu time required to rasterise each partition of the buffer increased only slightly to 11.5 seconds, although the reduction in the number of partitions generated a forty-four per cent increase in speed for the total map sheet to 288.244 seconds.

There is much evidence to suggest that careful tuning of the software is essential to achieve reasonable use of resources and of processing rates. The effect of inappropriately ordered input vectors is critical in slowing the system down; to read the full vector file for each partition of the raster buffer is manifestly inefficient since over ninety per cent of the data are often rejected for each strip. Franklin (1979) has itemised a series of options available to restructure vector data for raster partitioning. Two of his suggestions were tested for relevance in the rasterising of Ordnance Survey data: these were his algorithm 2 (rewrite rejected vectors into a secondary file) and algorithm 3 (sort the input vectors by strips first). A summary of the effects of the software and data tuning is given in Table 5.5

Table 5.5: The effects of software tuning on the vector
to raster conversion of OS 1:1250 data

Map sheet: NZ 2642NE Number of x,y points = 14,789

Raster image: 5000 x 5000 elements

Pixel resolution: 0.1 ground metres

Input data	Storage per pixel (bytes)	No. of strip partitions	Total storage (Mbytes)	Total cpu time for job (secs)	Comments
Unsorted vectors	2	50	50	525	Initial tests
Unsorted vectors	1	50	25	510	Pixel data compaction
Unsorted vectors	1	25	25	278	Increase in partition size
Unsorted vectors with rewrite to secondary file	1	25	25	227	Much use of backing store for vector data
Sorted vectors	1	25	25	243	Includes 64 seconds for vector pre-sort

5.4 DATA COMPACTION TECHNIQUES FOR THE RASTER IMAGE

It is unreasonable to use twenty-five megabytes for the storage of one 1:1250 raster image; even with careful blocking techniques, such large volumes of data will absorb in excess of sixty per cent of a standard 1600 bpi magnetic tape. This would imply a 300-fold increase in storage of the national digital topographic data base as compared with the unpacked vector format described in section 2.4.2. When allied with degrees of sparsity of ninety-seven per cent or more, all this suggests that exploration of efficient data packing techniques is essential. In the design of effective compression methods, an effective balance has to be adopted to ensure that benefits gained are not impeded by lengthy central processor operations to compact and to expand the data. Thus, the use of the data in their expanded form ensures straight-forward FORTRAN coding at the software input and output stages. Under the MTS operating system, a single READ or WRITE operation involving the transfer of a one-megabyte strip of the raster image has no dramatic effects on software performance when compared with transferring the same data in packed form; this transfer operation takes three milliseconds of central processor time for unpacked data and two-and-a-half milliseconds for packed data. It is the reduction in backing store requirements which will be the most significant benefit from any

compaction attempts.

The handling of large sparse matrices in a computer system is not a recent development; Pooch and Neider (1973) give a useful summary of indexing techniques for mathematical applications. More recently, Visvalingam (1976) has investigated some of these ideas for the efficient storage of the 1971 GB population census data (see section 3.5.1). These methods of compression are not entirely suited to picture processing applications (Cederberg 1979) since many of the relational links between nearest neighbouring pixels - especially across scan-lines - are lost. In essence, it is necessary to adopt a hybrid strategy which is suited to both archiving and application requirements. In practice, modern computing equipment (e.g. terminals and raster displays) is gradually alleviating this need to make trade offs between data storage and display; Allen et al (1980) describe the developments of intelligent microprocessor-controlled terminals and displays which remove the burden of data reformatting from the main computer for rapid and easy interfaces between the user and the machine.

A series of data compaction methods have been investigated for the Ordnance Survey data involving such techniques as the address-map scheme, the row-column scheme and the threaded list scheme (see Pooch and Neider 1973). Generally, though,

only the bit-map scheme (Pooch and Neider 1973) and run-length-encoding (Rosenfeld 1969) were expected to have direct relevance since these strategies suit the short-runs of connected set pixels inherent in each raster scan of the large scales data. An initial assessment of the performance of the bit-map scheme can be made immediately for a particular sheet once some figures for the degree of sparseness is known (see section 3.4.1 and Table 3.4), although run-length-encoding methods are dependant on the nature of each scan-line. Section 3.4.1 shows how the characteristics of the large scales map data do not fit neatly into run-lengths as would be the case with the smaller scales. A modification to these ideas - the delta-indexing scheme (after Visvalingam) - is considered to be highly appropriate for the compression of OS monochrome data. A full description of this and several other techniques are described in section 3.4.1. Robertson (1980 p 109) - in his M.Sc. dissertation describing the development of a scanning system to handle 'simple' topographic maps - has suggested that a coded delta like method of compaction yields an expansion rather than a compression of his raster storage for contours, culture and drainage data. This contradicts the results of investigations made in this research; the technique seems highly suited to OS large scales data characteristics. This apparant difference of opinion probably has arisen because Robertson is discussing the

effects of bit-sequential scan-line data direct from a scanning device in which lines have not been thinned to one-pixel widths. The effects of this distinction are unknown at present.

5.4.1 Development of a coded-delta compaction algorithm

The coded-delta method of raster compression involves transforming the long runs of zero elements in each scan line into run-lengths to separate the non-zero pixels. The distinction between these run-lengths and attribute codes is ensured by negating the run-length values before storage. Figure 5.6 below shows the design of a coded-delta scheme for the compaction of an OS large scales raster scan line.

Figure 5.6: Example of a coded-delta compression scheme

001000000000000100000010000010000000000000000000010000

becomes

-2, 1, -13, 1, -6, 1, -5, 1 -17, 1, -3

In reducing the storage overheads associated with the uncompact raster images, it is necessary to run length encode each scan line prior to storage. The variable lengths of each scan-line record after compression presents problems in the data structure and the input and output stages of the software. Since the record lengths are no longer fixed, then straightforward unformatted FORTRAN block transfers of arrays to and from backing store are not possible; this technique is extremely rapid since, in an IBM implementation, it necessitates only one call to the IBCOM subroutine rather than one call for each element of the array. With such high data volumes careful control of I/O methods is crucial. Two alternatives are available:

- A header section is attached to each scan line giving details of its length (the access vector ideas described in figure 3.2a). In this way, implied DO-loops can be used with each READ and WRITE statement to control the length of each record which is transferred to the data file. This technique has been rejected because:

(a) additional storage is necessary for each record header, which increases data volumes and the complexity of the data structure;

(b) the use of implied DO-loops in FORTRAN I/O is very

slow (see Table 5.7) - hence, this is undesirable with many data transfers.

- Most operating systems provide a low-level method of data transfer known as executive read and writes. Their function is to transfer a specified number of bytes from a defined address in the central processor memory to backing store. No header information is included and the method is very fast since no validation of the data transfer is performed.

Table 5.7 compares the cpu time required on the IBM 370/168 to read in the total 5000 x 5000 raster image by the three described methods. These investigations indicate that executive reads and writes are faster than FORTRAN block transfers, at least on the IBM 370/168 running under MTS.

Table 5.7: Comparison of the central processor times
required to read a complete raster image of
5000 x 5000 pixels scan line by scan line

(N.B. SCAN is an array of 5000 elements, each of one byte)

<u>Method</u>	<u>Hypothetical</u> <u>code</u>	<u>370/168</u> <u>cpu time</u> <u>(secs)</u>
FORTTRAN block read	DO 1 I=1,5000 1 READ(1) SCAN	13.0
FORTTRAN implied DO-loop	DO 1 I=1,5000 1 READ(1)(SCAN(J),J=1,5000)	175.0
Executive I/O transfer	DO 1 I=1,5000 1 CALL READ(SCAN,5000)	10.1

The allocation of one byte per pixel within the raster buffer necessitates a maximum run-length of zero elements (the coded deltas) to be 127: larger values than this cannot be stored in eight-bit units. It is, however, perfectly feasible to place a series of coded deltas consecutively in each scan line image. For example, a record comprising the following pixel (numerical) sequence:

30, -56, 6, -127, -127, -35, 22, -6

represents the following uncompact scan-line data

FC 30, 56 zeros, FC 6, 289 zeros, FC 22, 6 zeros

where FC = feature code

Two serial-based FORTRAN subroutines were written to compact each scan line into a coded-delta format for storage and expand it back into raster scan notation for any post hoc processing and display. The source code can be found in Annex 5d: SUBROUTINE CODER takes each raster partition and compacts it for storage and SUBROUTINE BUFF reads in the compacted scan data and unpacks it.

The efficiency of any data compression technique is dependant on the characteristics and complexity of the source data; a map sheet with a high-frequency of linear features will not compress to such an extent as would be the case with a sheet of relatively sparse cover. The degrees of compaction possible with the coded-delta scheme are closely allied to the orientation of the linear features. A hypothetical map sheet comprising all horizontal lines will

Table 5.8: The effects of data compaction
on two O.S. 1:1250 map sheets

Description	Map Sheet		Units
	NZ 2641NE	NZ 2642NE	
No. of vector points	5005	14789	
Total length of line work at plot-scale	14.7	39.8	metres
Size of vector data file (in DMC format)	70224	248920	bytes
Estimated vector storage if compressed	18000	63000	bytes
CPU time to convert to raster (no compaction)	147.2	278.1	seconds
Size of raster image with 0.1m resolution	25,000,000	25,000,000	bytes
Degree of sparsity	99.2	97.8	percent
CPU time to convert to raster (compacted)	175.2	354.0	seconds
Increase in CPU time with compaction	19	27	percent
Size of raster image after compaction	469,297	956,455	bytes
Degree of compaction	53.3	26.1	times
Size of compacted raster image in relation to compacted vector file	26	15	times
Size of compacted raster image in relation to DMC format vector file	6.7	3.8	times

not compact to the same extent as is possible with all vertical lines, assuming that all the run-lengths were arranged horizontally.

The performance of the compaction algorithm has been encouraging. Table 5.8 shows the results of the compaction process on two dissimilar 1:1250 map sheets; a comparison with the uncompact raster images is made.

The evidence described in table 5.8 suggests the performance of any raster-based manipulation of data on a serial-machine is closely allied with the characteristics and volume of the vector data. This is not so much the case with parallel algorithms (see chapter six). A twenty seven per cent increase in the central processor time provided a twenty six times reduction in the storage overheads for the dense sheet; far better results were gained for the sparsely covered sheet with figures of 19 per cent and 53 times respectively. To some (unknown) extent this will be due to the maximum compaction rates possible within O.S. raster systems. Even after compaction the size of the grid-based data files are 3.8 and 6.7 times larger than the equivalent unpacked vector formats for the dense and sparse data respectively.

The rates of compression achieved by these methods compare favourably with those reported by other research workers in the field. Robertson (1979) suggests the size of his compacted raster files to remain 3 or 4 times larger than his equivalent vector files. Pferd and White (1981) report a five times reduction in the compaction of raster images of engineering drawings using run-length-encoding techniques. The Bell Laboratories scanning system that they describe (see section 7.3.2) converts from raster to vector in real time and hence the compaction scheme and resulting packing format for the raster data is kept simple to enable rapid and straight-forward conversion to vectors in stages which follow.

5.4.2 Software tuning

An investigation into the degrees of compaction for each scan line in the raster image of NZ 2642NE has discovered that the lengths of the resultant records (scan-lines) are strongly correlated with the inverse of the frequency of zero pixels ($-0.93 r$). A summary of the characteristics of the compressed scan lines is given in Table 5.9.

Table 5.9 A summary of the characteristics of each scan-line
for sheet NZ 2642NE after coded-delta compaction

Pixel resolution 0.1 ground metres

Description	No. of bytes	No. of zeros	No. of set pixels	No. of coded-deltas
Max. record length	369	4723	277 (5.5%)	92
Mean record length	223	4857	143 (2.9%)	80
Min. record length	78	4957	43 (0.9%)	35

Since a ninety three per cent r^2 correlation exists between the frequency of zero pixels and the resultant record lengths in each scan line, then the remaining seven per cent is taken up by the unknown degree of contiguous coded-delta elements (i.e. runs which are greater in length than 127 pixels). This factor is dependant on the characteristics of each map image; an investigation was made into the effects of extended coded-delta lengths. The hypothesis developed was that the use of two-bytes per element after compaction would be more efficient since the chance of contiguous coded-delta values would be nil. A 16 bit element provides for the storage of any integer in the range -32768 to +32767. The

representation of each set pixel (i.e. those with attribute codes) in two-byte units would waste space: table 5.9 shows that on average only 2.9 per cent of each scan line has set pixels (in a 5000 x 5000 pixel image for this test sheet), and hence the redundant space will be minimal. These tests proved the hypothesis to be wrong. Investigations into the use of two byte coded-deltas have produced compacted raster files that are almost twice the size of the one-byte method. The size of the compressed 5000 x 5000 image of NZ 2642NE rose to 1.7 megabytes (a 1.8 times increase). Further studies have shown this to be the case with raster files of various sizes and complexity. The benefits of removing contiguous coded-delta elements do not outweigh excessive space taken for the set pixels and relatively short runs of zero-elements.

.5 THE CHOICE OF PIXEL RESOLUTION

The resolution of each pixel in describing a raster image has considerable importance. Careful choice of this cell-size is crucial; it dictates the time necessary to scan the map sheets, the degrees of exactness of the digital representation, and the quality and appearance of any output graphics. This does not suggest that the choice of resolution at the input (scanning) stage should necessarily

decrease the characteristics of the resultant displays or digital manipulation, although caution should be exercised in any attempted scale changes. Clearly, though, the precision of any raster image can be no better than the weakest link in the system and, in this project, output quality can be generally no better than the accuracy of the input document. It is possible to produce simulated higher quality by regriding input data on a higher resolution raster and then skeletonising the results - i.e. thinning linework to one pixel width.

A decision on the pixel size to be adopted has an exponential relationship with uncompact size of the raster files; this has serious implications on data storage and central processor time. In these circumstances, it is futile to develop and process raster images with a precision that is higher than required for analytical work and display. Conversely, raster files with insufficient resolution cannot be used with any reliability for high-precision work.

The effects of pixel sizes on the characteristics and handling techniques of topographic data in a raster notation have been investigated empirically. The development of the serial-based algorithm for vector-to-raster conversion (section 5.3.4) was designed to facilitate user-defined pixel sizes; this information is supplied to the software as a

series of parameters, along with the input vector file. A succession of conversions to raster were made on the Durham 1:1250 sheet (NZ 2642NE) in which pixel sizes were amended to range between 0.1 metres (0.08mm) and 5.0 metres (4.0mm). The effects on process times, file sizes, degrees of sparsity were monitored in addition to the performance of the coded-delta compaction technique (section 5.4.1). The extent to which a raster file has to be partitioned during the conversion process has severe effects on cpu times since it requires many passes through the vector file (one pass for each partition). As far as was possible, the number of partitions was kept small although no more than one megabyte of main memory was used in each run. The results of these tests are summarised in Table 5.10.

The results provide conclusive evidence of the degradation which increasing pixel resolution causes on the rate of processing on a serial processor. The central processor time increases with the number of pixels describing the image; the degree of increase is highly dependant on the number of passes through the vector file needed at the input stage. Figure 5.11 shows the effects graphically. The rapid increase in the processing time can be seen to occur with pixel resolutions of 0.5 metres and less: the high number of partitions necessary to store the large volumes of data account for this. There is only a marginal rise in

Table 10: The effects of pixel resolution on the raster image of sheet NZ 2642NE

Pixel Ground Resolution (metres)	Raster Dimensions (pixels)	Size of Raster Partition (pixels)	No. of Passes	Size of Raster file (Kbytes)	Percentage Unset pixels	Size of file IEM 370/168-Process time	
						after compaction (Kbytes)	without compaction (secs)
5.000	100 x 100	100x100	1	10.0	37.14	7.36	1.57
2.500	200 x 200	200x200	1	40.0	57.66	22.11	1.81
1.250	400 x 400	400x400	1	160.0	75.12	57.20	2.32
1.000	500 x 500	500x500	1	250.0	79.51	75.65	2.65
0.800	625 x 625	625x625	1	390.6	83.35	98.88	3.12
0.625	800 x 800	800x800	1	640.0	86.82	131.16	3.83
0.500	1000x1000	1000x1000	1	1000.0	89.39	167.49	4.87
0.400	1250x1250	1250 x 625	2	1562.5	92.88	215.46	9.72
0.250	2000x2000	2000 x 400	5	4000.0	94.63	349.16	31.60
*0.160	3125x3125	3125 x 125	25	9765.6	96.54	564.49	168.36
0.125	4000x4000	4000 x 200	20	16000.0	97.29	740.45	185.71
0.100	5000x5000	5000 x 200	25	25000.0	97.83	956.46	278.13

* A special case involving 25 strips (passes) to ensure that each strip is completely filled with map data: ie there is no waste of space. This case served as a test for the effects of inefficient raster partitioning.

THE EFFECT OF PIXEL SIZE ON COMPACTION TIMES

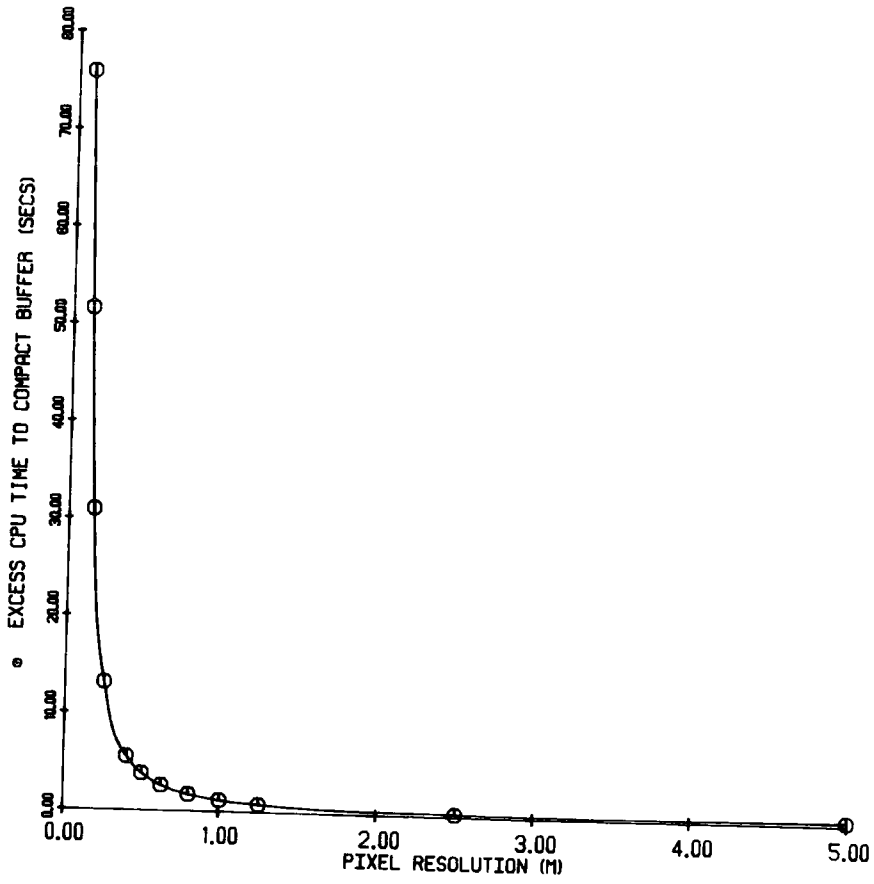


Figure 5.11

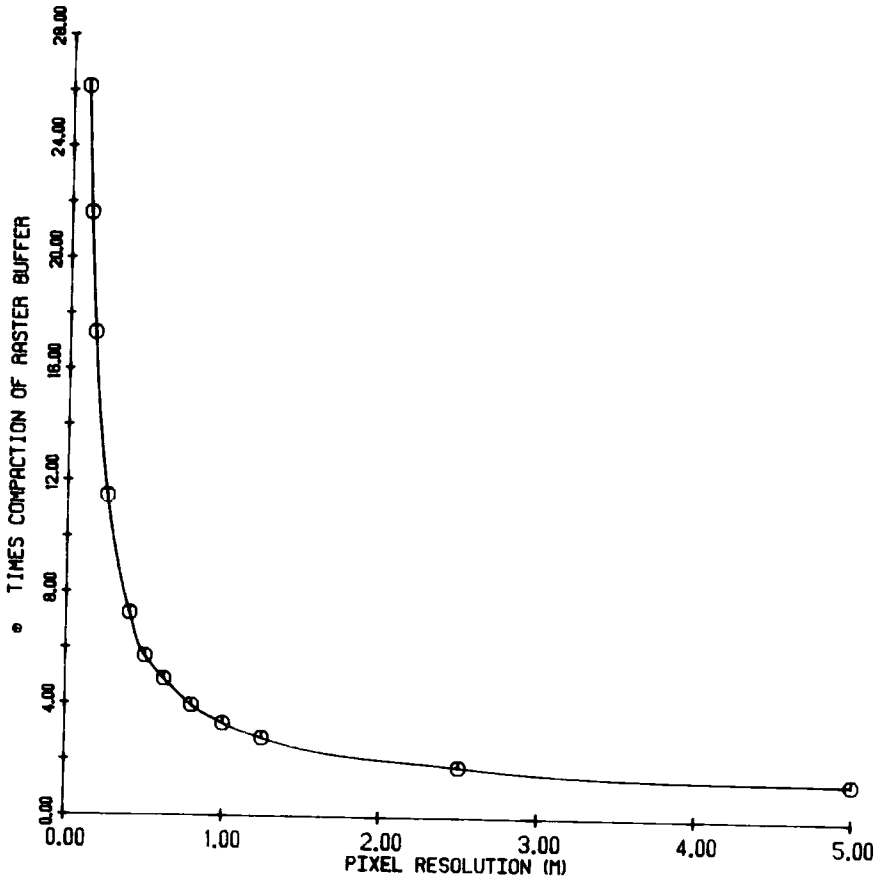
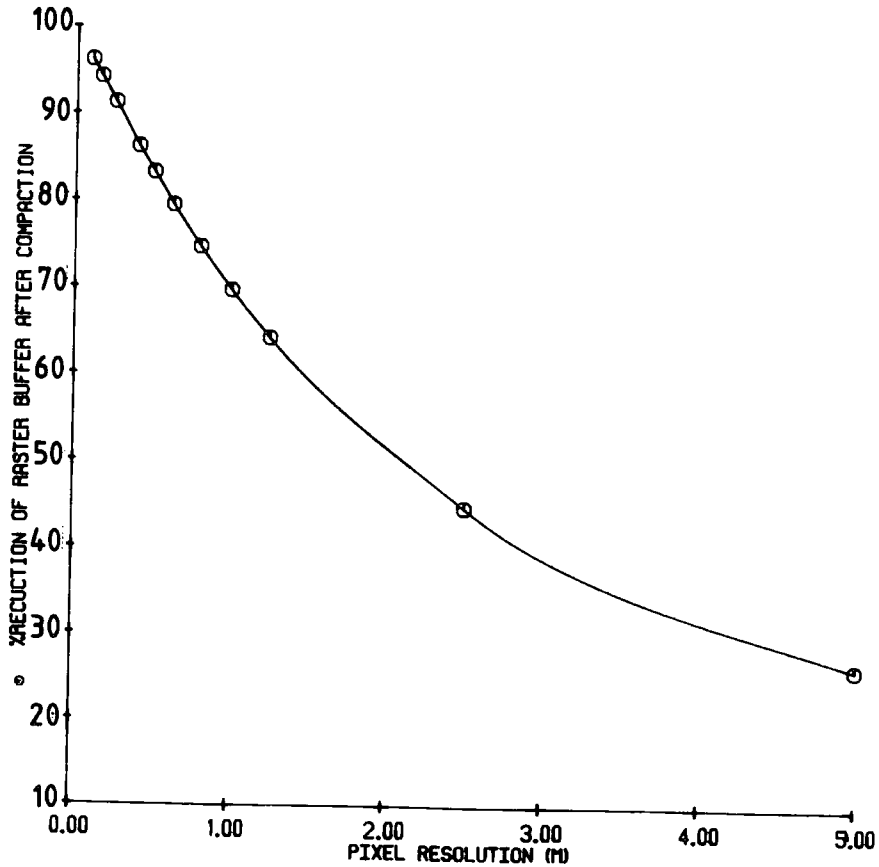


Figure 5.12

Figure 5.13

PERCENTAGE REDUCTION OF RASTER IMAGE WITH PIXEL SIZE



THE EFFECT OF PIXEL SIZE ON PERCENTAGE SPARSITY

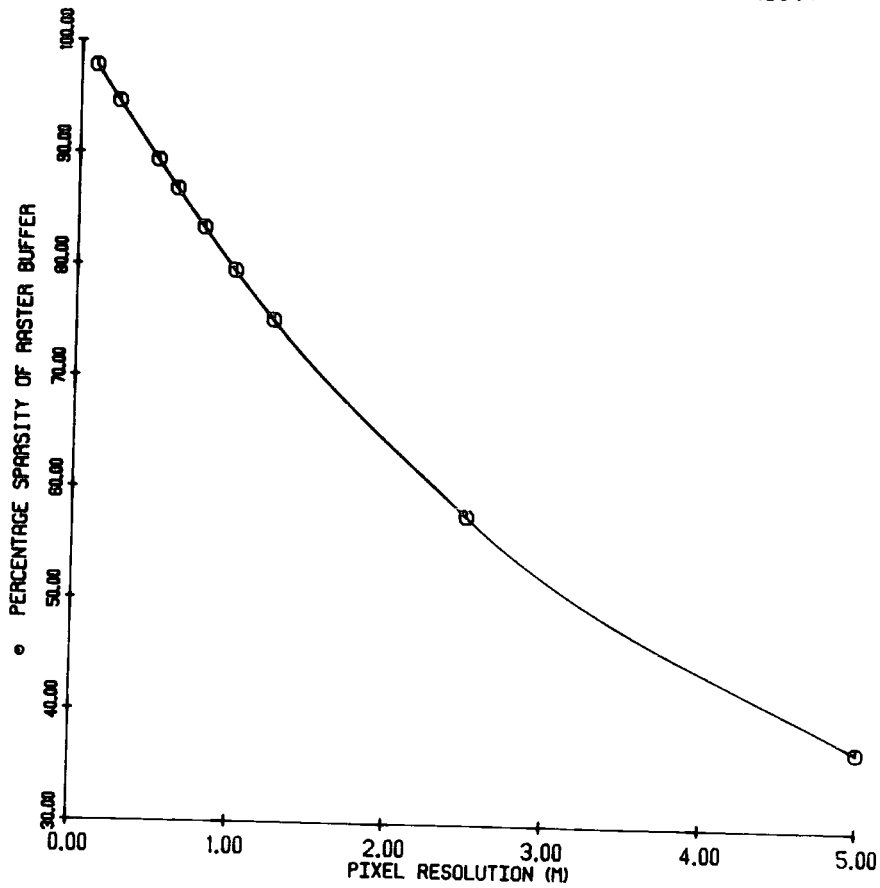
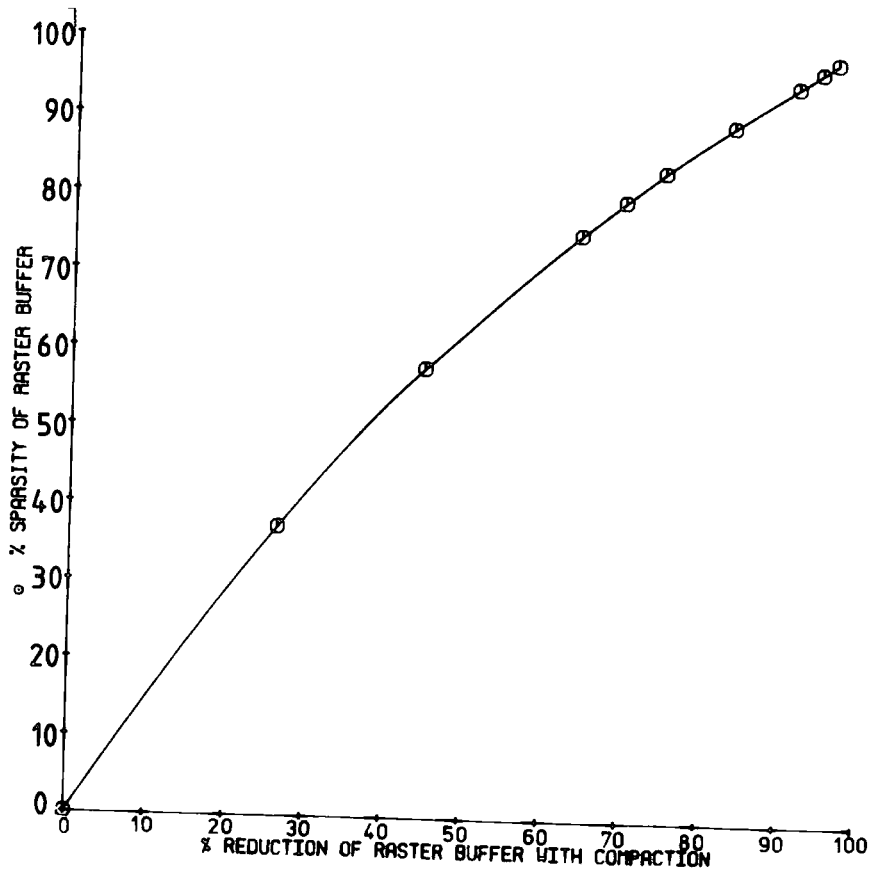


Figure 5.14

Figure 5.15

PERCENTAGE SPARSITY AGAINST PERCENTAGE REDUCTION OF RASTER BUFFER



conversion speeds with pixel resolutions where the complete task can be handled with one data partition.

There is similar evidence to suggest that the performance of the data compression methods are closely allied with pixel size: figures 5.12 and 5.13 describe the results. Both figures show similar measures of performance: figure 5.12 describes the degree of file compaction with changing pixel resolution and figure 5.13 displays the same trend on a percentage reduction basis. These trends are encouraging. The degrees of data compression increase quite dramatically as the pixel size becomes smaller; furthermore the rise in the size of the compressed data files are nothing like as rapid as the rise in the size of the uncompact buffer sizes. If O.S. sheet NZ 2642NE is assumed to be fairly typical of the 1:1250 data sets investigated and - hopefully - of the national situation, then there are valid reasons to expect a compaction of between ninety-five and ninety-nine per cent of the unpacked data with pixel resolutions adequate for Ordnance Survey standards. The measures of compaction performance reflect the degrees of empty space (sparsity) in the raster images. It is to be expected that an increase in pixel resolution will increase the frequency of zero pixels: figure 5.14 shows this trend. The evidence suggests a steady increase in percentage sparsity with finer (and hence a greater number of) pixels. The increases in the measures of

sparseness and the performance of the data compression in the raster image with smaller pixel resolution are not equal. Evidence from the figures shown in Table 5.10 suggests a marginal improvement in the rate of removal of empty pixels as their relative frequency increases. This trend is shown graphically in figure 5.15: the steady reduction in the gradient of the line is a measure of the slight increase in the performance of the coded-delta method of compression. The extent to which this is dependent on the characteristics of the map sheets is unknown.

In conclusion, these results quantify the intuitive expectations of raster data handling, at least on a conventional computer. An increase in the size of the raster image requires an extensive increase in computing resources. The times required for data manipulation increase exponentially with a reduction of pixel resolution; in contrast, there is evidence (see Chapter six) to suggest that this increase is nearer to linear on a parallel processor. These results stress the high diversity of performance rates in raster data handling; any quoted serial-processor times are only relevant if associated details about the image are supplied, such as pixel dimensions and degree of sparseness.

The ability to compact the data for storage purposes appears to be highly promising. The rate of increase in the

volumes of the compacted data with increasing pixel resolution is far less than as in the expanded raster file. Hence, the evidence suggests that it is the processing times which degrade any compaction technique and limit the resolution which can realistically be utilised. Figure 5.11 shows the effects of increased processing times to compress images with increased resolution. These rates of increase reach alarming proportions with high resolution pixels; this is a clear indication of the inappropriate architecture of a serial computer for these applications. There are two courses open to alleviate these problems:

- Wherever possible, expanding a raster image once it has been compressed should be avoided; this seems to be strongly supported by the Institut Geographique National, Belgium (Loodts 1981), which is undertaking intensive research into raster methods on serial processors.
- There is a strong case for the serious investigation and development of array processing algorithms for true parallel processors. Assessments made by this author suggest there to be much potential in this field: these results are described in chapter six.

5.6 WORK APPLICATIONS RELATED TO OS REQUIREMENTS

Section 4.2.2 itemises several of the rudimentary manipulative tasks which favour raster methods of data organisation. In particular - in the light of related experimentation - it has become clear that raster data processing can provide an efficient alternative to the vector strategy described in section 2.7 for the detection of links and nodal points in digital map line data. Consideration of this is given in section 5.6.1.

Another application which has serious implications for future raster strategies within the OS is efficient raster-to-vector procedures. Detailed consideration of the commercial solutions to this problem is given in chapter seven, although tests have also been carried-out in this research to examine the quality and procedures necessary to rebuild the manually created raster files discussed in this chapter back into a vector form. The results of such an exercise provides a means of appraising the raster-to-vector process when presented with error-free, manually created test data. The details of this exercise are reported in section 5.6.2.

5.6.1 Some applications toward link and node detection

There is much evidence from the research to suggest that the restructuring of Ordnance Survey vector data is far more straight forward with a raster strategy - an approach which has hitherto been ignored by the project team in the national mapping organisation (see Thompson 1978).

Section 2.7 describes the OS restructuring project which aims to generate vector data with much topological structure. The process involves reformatting the existing data notation into a series of low-level links and nodes. In principle, the identification of such node positions during a vector-to-raster conversion is manifestly simple; in contrast, it has proved to be costly and slow with the OS vector-based strategy.

Investigations have shown the raster-based concepts for node and link extraction to be simple to implement. As the vector to raster conversion is taking place, the software can check to see if each pixel has already been assigned a feature code. If the test proves to be positive, then the pixel location must be a node position. Each time a node is detected, its address can be appended to a node register within the computer system. Tests performed on these notions have proven to be favourable although a series of

reservations do exist:

1. The design of the vector-to-raster algorithm (section 5.3) in its present form is inappropriate for such an application. The equation for each vector is solved first in the x direction and then in y to ensure pixel continuity along each line. As a consequence, feature codes are frequently, but not invariably, assigned twice to the same pixel - once in each pass for x and y - and hence the software can become confused when searching for nodal points. Two solutions have been considered:

(a) the incorporation of feature serial numbers with the attribute values can distinguish each linear entity in the raster image although the current storage conventions do not allow this strategy;

(b) pixels are not transferred to the raster buffer until the vector has been processed completely. This method works well although certain vectors are sufficiently long to cause serious storage problems in a temporary buffer.

2. Current specifications of the Ordnance Survey data are not adequate for such digital manipulation. For example, polygon features (buildings, etc.) have not been modified

to ensure that start and end points are coincident. For these reasons, some node locations can be missed completely. The strategy adopted in the Ordnance Survey restructuring project has involved the introduction of complex positional tolerances to distinguish point coincidence. Clearly any raster alternative must be closely allied with pixel resolution.

In short, these tests have been promising. The raster method of detecting nodes has identified sixty eight per cent of all nodes in a test map-sheet. The tests have attempted no method of controlling the problems discussed in point 2 above. This will require careful tuning and modifications to the software; similar amendments have been lengthy in the vector-based approach by the Ordnance Survey.

.6.2 Attempts to rebuild the raster data image into a vector file

Most manufacturers of scanning equipment offer options to convert the raster data into vector strings for conventional processing, data-tagging and easier interactive validation. Chapter seven points out how most of these software options are piecemeal, providing ad-hoc line-work with an inadequate

structure for much digital manipulation. Generally, the quality of the derived vectors is not only dependant on the design of the conversion algorithm but also on the quality of the raster data provided at the input stage. Clear examples of these factors are given in figures 7.6 to 7.14 (see chapter seven). The raster-images which have been generated directly from the Ordnance Survey digital vector files (described in section 5.3.4) have provided a datum with which to compare the results achieved by scanners and post-hoc vectorising. Tests have been carried-out with the kind cooperation of the Image Analysis Group at Oxford University to convert these simulated raster images back into vector-form. The process involves thinning each linear feature (in raster) to a single pixel width, linking them together and then filtering-out superfluous points (see figure 6.13).

The Oxford method of raster-to-vector conversion (outlined in Harris et al 1981) is one of the few known procedures which has attempted to incorporate some topology in the resulting vector data structure. This is provided by the detection of each node (and associated segments or links) in the resulting network of lines. The generation of separate node and link files - with appropriate cross-referencing between the two - provides sufficient structure for entity rebuilding in later stages. The Oxford software has been

tuned towards the handling of polygon data to suit the needs of digitising in the Department of the Environment (see Davey et al 1977). The resultant output is a file of coordinates representing polygon boundaries in vector-form with their associated feature-types printed along-side them. The polygon recognition process is achieved by amalgamating the appropriate nodal-points with their associated segment data. Feature-code assignment is accomplished by interactive means once the polygon files have been constructed.

Clearly the data structure derived from the software is not directly comparable with the OS large scales data in which certain features are inherently linear, although there are a number of similarities with the Restructuring Project discussed in sections 2.7 and 5.6.1. Attempts were made to rebuild a simulated raster-image using the Oxford facilities; figure 5.16 shows a section of the vector-file after conversion from raster. The results are encouraging; there is no evidence of feature omissions and the connection of points is accurate and relatively smooth. There are instances where the stepping effect of the line-work (associated with raster images) is evident although this is restricted to lines nearing a north-easterly and north-westerly direction and is a function of the pixel resolution adopted (0.1mm at 1:1250 scale).

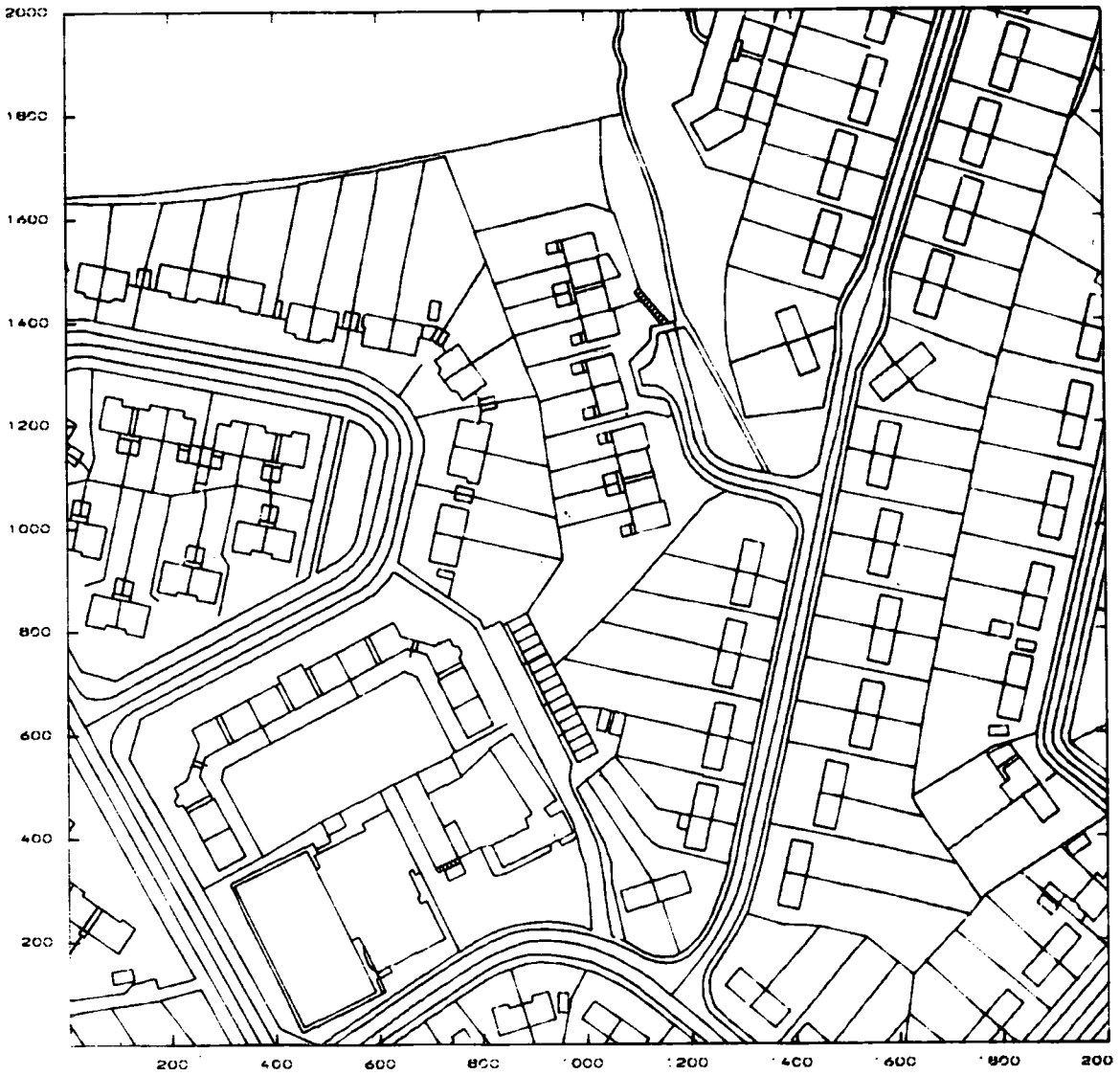


Figure 5.16 Section of a vector file
rebuilt from a simulated raster image
of OS sheet SO 9283SE.
Crown Copyright Reserved.

There has been little necessity to thin these raster lines (to single pixel widths) since the digital vector-to-raster software already provides the data in an appropriate form i.e. one pixel wide; the Oxford software need only validate the pixel data and logically chain them together. This factor will have enhanced the quality of the results considerably since it is the line thinning stage which is most arduous (Boyle 1979). The cpu times for the conversion on the Oxford DEC PDP10 computer are given in table 5.17:

Table 5.17: DEC PDP10 process times to convert 4096 x 4096 raster image of 1:1250 sheet SO 9283SE into vector form.

Pixel input	142 seconds
Skeletonising	335 seconds
Node and segment formation	408 seconds
Output	70 seconds

Clearly the vectorising stage is a lengthy process - excluding the time taken for data input and output, the conversion has taken in excess of 740 cpu seconds. The considerable times associated with the data I/O are allied with the volumes involved (Franklin 1979) and these account a further 22 per cent of the total time. It is difficult to compare the processing times quoted here with those given earlier describing work carried out on the IBM 370/168 at

NUMAC; it is likely that the IBM machine is 2-3-4 times as fast as a DEC 10 computer (depending on the model). Such large processing times for a relatively low pixel resolution and line-work which is already skeletonised (thinned) is intolerable for a production system. An increase of c. forty per cent in the processing times is to be expected with data collected directly from a scanning device; extensive line thinning and validation will be required before any pixel-chaining and node and segment identification can begin. The extent to which these times are dependant on inappropriate software-tuning is unknown; certainly, more favourable speeds may be attained with software which is designed to suit the characteristics of the input data more closely. The work of the Oxford team is admirable given the volume of data involved and complexity of the Ordnance Survey polygons. The quality of the resultant vectors and the characteristics of the data structure has potential in future digital representations of topographic information; Ordnance Survey has monitored a link and node strategy since the mid-seventies (Thompson 1978). However, evidence from these tests show that the existing processing rates for post hoc manipulation of scanned data are inappropriate in a commercial environment. These rates are largely a result of the inappropriate design of existing computer systems and algorithms.

For the complete map-image, the Oxford raster-to-vector program produced 4671 segments and 3440 nodes: it would have been instructive to compare this summary with equivalent information from the Ordnance Survey restructuring software. This is not possible, however, since the PMA schema (See section 2.7.1) introduces additional node-points at each 100 metre grid square boundary. Further, it is not the current practice for the Ordnance Survey restructuring software to provide a summary of the total number of nodes and links describing a map-image.

Generally, the description of a nodal-point is speculative, particularly in the Oxford vectorising software. The node-count of 3440 for the test 1:1250 data comprises nodes having two or more segments attached. An example of a two-order-node occurs with closed, isolated building polygons - where two lines will enter one node in that feature. Post hoc software can remove two-order-nodes if required: this has taken an extra 110 cpu seconds and results in 2730 nodes and 3996 segments. In this way, it is possible to reduce the complexity of the digital map data since little over twenty per cent of the original node-count comprises two-order-nodes.

5.7 CONCLUSIONS

The investigations described in this chapter on the generation of raster images from digital vector files of topographic data and their subsequent packing and unpacking have served several purposes:

1. They have given an insight into the techniques and problems associated with handling large arrays of image data in a serial processing environment.
2. It has been possible to quantify the extent of computer resources necessary - at least for the vector to raster conversion.
3. Error-free raster data sets have been developed which have facilitated many ancilliary investigations:
 - (a) comparisons with raster images obtained from true scanning sources;
 - (b) essential statistical analyses to determine characteristics and contents of the data;
 - (c) tests on the quality and precision of raster-based display devices;

(d) input data known to be 'clean' have been made available to study the feasibility of array processing techniques and to discover the types of manipulation possible on these new generation machines, including appraisals of raster-to-vector conversion software.

It has not been the intention of this work to develop a robust, highly advanced method of producing raster files from OS vector-based customer data sets. Indeed this work has made extensive use of the facilities provided on a powerful mainframe computer; comparative software for smaller mini-computer installations will necessitate further amendments and tuning to be made.

Graphically, at least, the results have provided a means of comparison with those achievable through conventional OS production methods and with the results possible by purely automatic means - these are described in detail in chapter seven.

The exercise has proven it to be technically possible to transform digital representations between vector and raster notations albeit with modified structure and much central-processor resources. There is evidence to suggest

that the problems associated with a raster-to-vector conversion for purely scanned data far outweigh those inherent in the conversion of simulated-images as has been the case here. Nevertheless, it is to be anticipated that fluent raster-to-vector-to-raster transformations will become necessary and more robust in the next few years.

APPENDIX 5.1

Determination of the equation of a straight line
knowing two points on that line

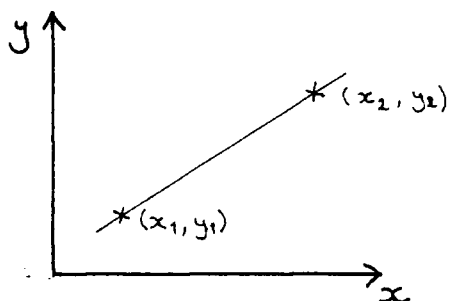
The general equation of a straight line is:

$$y = m * x + c \text{ -----(1)}$$

where m is the gradient

c is the value of y when x = 0

At least two points are required to describe a straight line:



Consider:

$$\text{gradient} = m = \frac{\text{diff. } y}{\text{diff } x} = \frac{y_2 - y_1}{x_2 - x_1} \text{ -----(2)}$$

Substituting (2) into (1) gives:

$$y = \frac{(y_2 - y_1)}{(x_2 - x_1)} * x + c \text{ -----(3)}$$

At point x_1, y_1 from (3):

$$y_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} * x_1 + c$$

To find c :

$$c = y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} * x_1 \text{ -----(4)}$$

Substituting (4) into (3) gives:

$$y = \frac{(y_2 - y_1)}{(x_2 - x_1)} * x + y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} * x_1$$

$$y - y_1 = \frac{(y_2 - y_1) * (x - x_1)}{(x_2 - x_1)}$$

Equation of a straight line knowing two points on that line is:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

=====

5 THE MANIPULATION OF RASTER DATA ON A PARALLEL COMPUTER

6.1 INTRODUCTION

Since the early seventies, a new generation of computing hardware involving array (or parallel) techniques has been undergoing stringent tests in an attempt to speed-up existing processing rates for some algorithms (Parkinson 1976). Not surprisingly, many of these tests have had their origins in the applied-sciences, where digital applications require vast amounts of central processor power: finite element analysis and modelling and simulation techniques in fields such as meteorology are all examples. There is little evidence of research of a similar nature for the handling of geographic data. By far the majority of known existing systems are based-around a serial-type computing environment (e.g. Brassel et al 1980, Calkins et al 1980, Cooke and Maxfield 1967, ESRI 1971, Harvard University 1974, MSDOA 1971, Peuquet et al 1980, Switzer 1975, Tomlinson et al 1976): a sequential method of digital manipulation has not restricted these developments unduly since the amounts of arithmetic operations tend to be limited in extent. The steady decrease in costs for computing hardware and the generation of advanced serial-programming techniques and data structures

and the concentration of vector data formats have been instrumental in preserving this conservatism.

Section 5.1 points-out how the increased data volumes and central processor dependance likely to arise with raster systems may well benefit from parallel processing techniques. The nature of the raster-based algorithms and the high demand for increased throughput of data is proving to be severely constrained by serial processors. The current design of these traditional machines appears to be nearly reaching the limits of its performance capability (see 5.2). Speed-up factors of the order of two to five may be achievable with the use of ^{very} large scale integration (VLSI) micro-circuits and heat extraction technologies although, even with these increases, there is evidence (see table 5.10) to suggest the need for serious investigation into the applicability of completely new approaches. The high-costs of hardware and the specialised software associated with these alternatives have not been conducive towards the development of faster and cost-effective raster systems. No commercially available raster-based systems (see chapter seven) have yet adopted a parallel-strategy of digital manipulation and, hence, any formal research into methodologies are limited in extent.

This chapter reports on the design and implementation of a series of tests carried-out to discover the feasibility and

performance of handling the Ordnance Survey large scales data in raster-mode on a parallel processor. There is much evidence to suggest that distinct benefits can be achieved with the adoption of these new generation machines which lend themselves readily to certain tasks far more favourably. It is crucial to develop them now as present indications suggest the widespread availability of these new hardware concepts within the next decade (Parkinson 1978).

6.2 AN ARRAY-TYPE COMPUTING ENVIRONMENT

A description of the architecture of conventional serial-computers has been given in section 5.2; the introduction of specialised array-processors has been a logical extension of these basic principles. The essence of an array-processor is a considerable "number-crunching" capacity; attempts have been made to increase the rates of central processor manipulation whilst little attention has been given to speeding-up the transfer of data to and from the backing store. Some fundamentally new approaches have been necessary to develop these computers which are orders of magnitude faster than the fastest sequential machines.

The separation of the main-memory and the arithmetic unit in normal computers has lead to a requirement for data to be

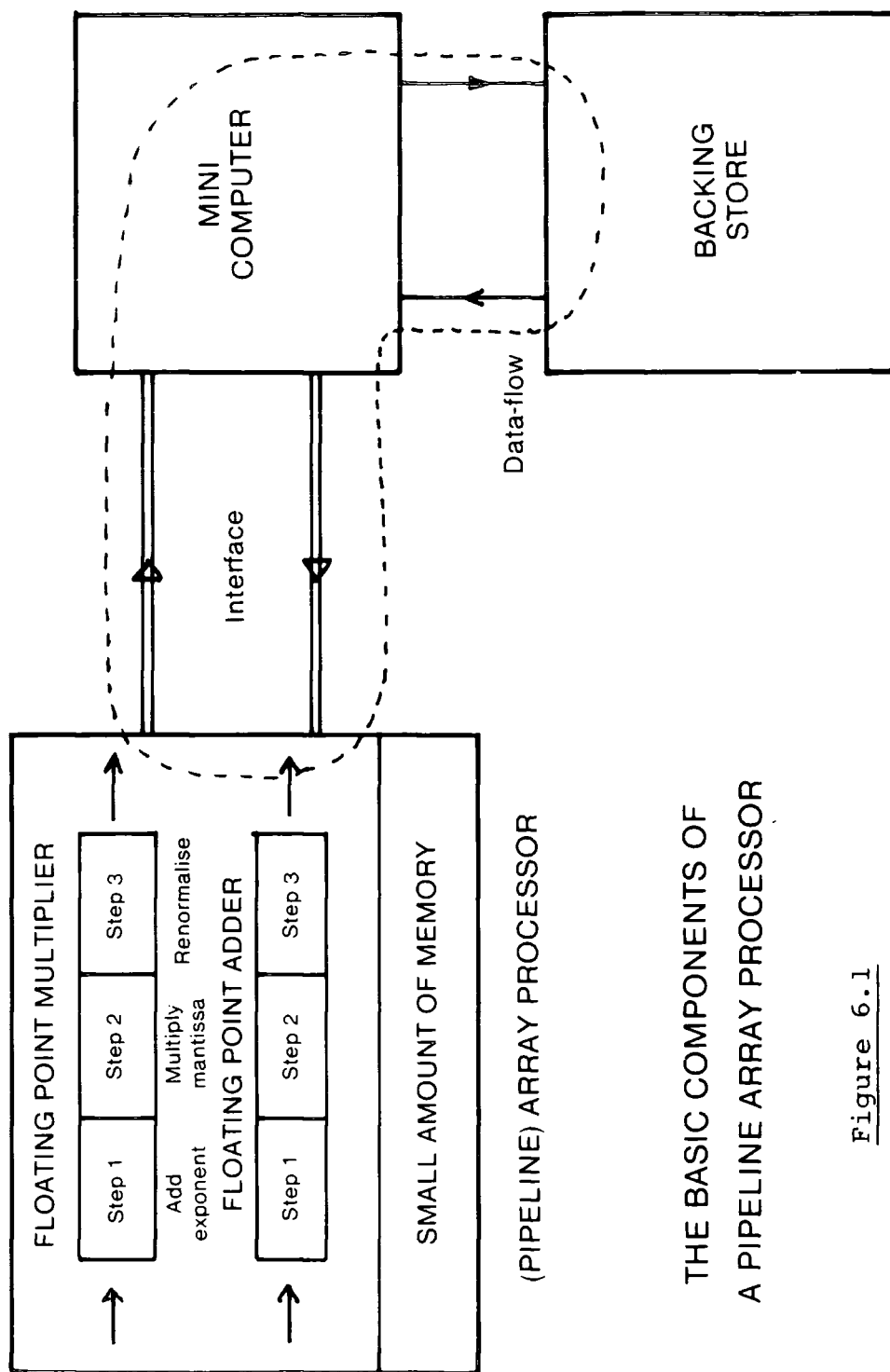
transmitted between the two; the speed of this data highway is an obvious bottleneck, particularly with voluminous raster data, and transfer rates many times those available from current technology are needed if dramatic increases in useable power are to be achieved. Since the modern sequential machines are now reaching the practicable limits of data transfer and processing speeds, attention has begun to focus on the design of these more specialised devices with the ability to provide multiple-processing streams, simultaneous data manipulation and so on. The term "array-processor" has come to be a popular description of this new-type of machine although it encompasses two very different types of architecture: an array of processors or an alternative design that is particularly efficient at processing arrays (the pipeline approach). Since confusion on the terminology of these different types of specialised hardware abounds, a distinction between the two will now be made.

6.2.1 The vector or pipeline array processor

The vector-array processor exploits the pipeline principle already used in many advanced sequential machines. The basic principle of the pipeline is to allow overlapping in the various stages of the decoding, data fetching and execution

of instructions. In this way, it is possible to perform more quickly providing that the algorithm is suited to the architecture and there are sufficient data to process. The devices emanating from Floating Point Systems Inc. have evolved around the pipeline principle: these are designed to connect as peripherals to existing computer systems to provide a specialised form of data processor for the right job. The AP120B can be connected to most popular mini-computers and the AP190L has a similar design strategy with interfaces appropriate for mainframe computers (Floating Point Systems 1979).

A simplified description of the elements in a pipeline array-processor is given in figure 6.1, in which the floating-point multiplier is used as an example. In binary form each floating-point multiplication comprises three operations: the exponents of the two numbers are added, their mantissas are multiplied and the result is renormalised into an appropriate form. With conventional sequential-machines, each pair of numbers must have completed all three stages before the next pair can enter the arithmetic unit. The pipeline strategy has been to allow the second pair of numbers to enter stage 1 as soon as the first pair have passed to stage 2. Thus, providing there are a sufficient number of numbers to multiply, the hardware will perform three times as fast as a conventional processor.



THE BASIC COMPONENTS OF
A PIPELINE ARRAY PROCESSOR

Figure 6.1

Similar design strategies are possible for the floating-point addition of numbers. Clearly there is an initial start-up time before the pipeline reaches its maximum speed of operation. To multiply one pair of numbers on an AP120B, for example, will take the same time as a serial machine - typically 500 nanoseconds. For a series of multiplications it is possible to reduce these rates by a factor of three: once the pipeline is full, a new answer is produced every 167 nanoseconds. The FPS brochure (Floating Point Systems 1979) quotes the possibility of twelve-million floating point computations per second; this, however, is dependant on the types of arithmetic to be performed and the volume of data involved.

Clearly these specialised machines have much potential for certain, high-computational tasks; they have proved to be highly useful for the Fourier transformations of large seismic data sets in the oil industry (Parkinson, pers comm 1979). Their suitability for the handling of geographic raster data has rather less potential, given the need to perform operations which, for example, may 'ripple outward' from one point in the array and, furthermore, there are a number of fundamental problems in the design strategy making the devices less-efficient than the theoretical hardware specifications lead us to believe. In general, the pipeline processor has only limited amounts of memory and must rely on

the main memory of the host computer for data storage. This necessitates the repetitive transfer of data items between the host machine and the processor, which is inclined to slow severely the computational rates: the vector processor will often spend most of its time idle, awaiting the supply of more numbers to "crunch".

The development of appropriate software is inclined to be arduous; the FPS array-processors are programmed in a low-level assembly language (APAL) or via a series of subroutine calls from standard high-level FORTRAN. In this way the users are restricted to those facilities offered by the manufacturer; these are difficult to change without detailed knowledge of low-level assembly programming and a thorough conception of the machine's architecture. Nevertheless, the specialised devices are proving to be favourable in a number of work environments: they are relatively inexpensive (c. 60,000 US dollars for AP120B) and provide powerful resources for intensive computing tasks. For example, favourable reports have emanated from the Royal Aircraft Establishment, Farnborough, UK who are using a PRIME 400 mini-computer/AP120B installation for efficient manipulation of remote-sensing data and image processing techniques (Hardy, Earwicker pers comm 1979). In RAE remote sensing applications, the sheer volume of the data is a major problem: the digital manipulation of each SEASAT (radar)

image has taken one full day to process with the pipeline processor. It is expected that this would have taken two-weeks by conventional methods.

Although Floating Point Systems Inc. are market leaders in the field of pipeline array-processors, they do not have a monopoly. Data General provides array-processing facilities by means of specialised options connected to their Eclipse S130 and S250 mini-computers. The Eclipse S130 with AP130 array-processor option costs c.40,000 US dollars for the 64Kbyte machine and c.55,000 US dollars for a 256Kbyte machine.

The IBM 3838 array-processor has not found its way into many applications outside seismic calculations and algorithms: the hardware is expensive (370,000 pounds sterling for a one-megabyte machine) and runs on all IBM mainframes supporting the Virtual Storage (VS) or (MVS) operating system with the exception of the System 370/135.

Some more extensively used vector-based array processors have been those of the Cray series of computer systems (Cray Research Inc. 1980). The first Cray machine was delivered in 1976 and the design strategy has since found its way into the following work areas: weather forecasting and climatology, petroleum research, structural analysis, nuclear research,

geophysics and seismic analyses, fluid dynamics, defence and medical research. Current costs of a Cray machine can range between five and ten million dollars, depending on the model of machine and extent of memory selected.

6.2.2 The Distributed or cellular Array Processor

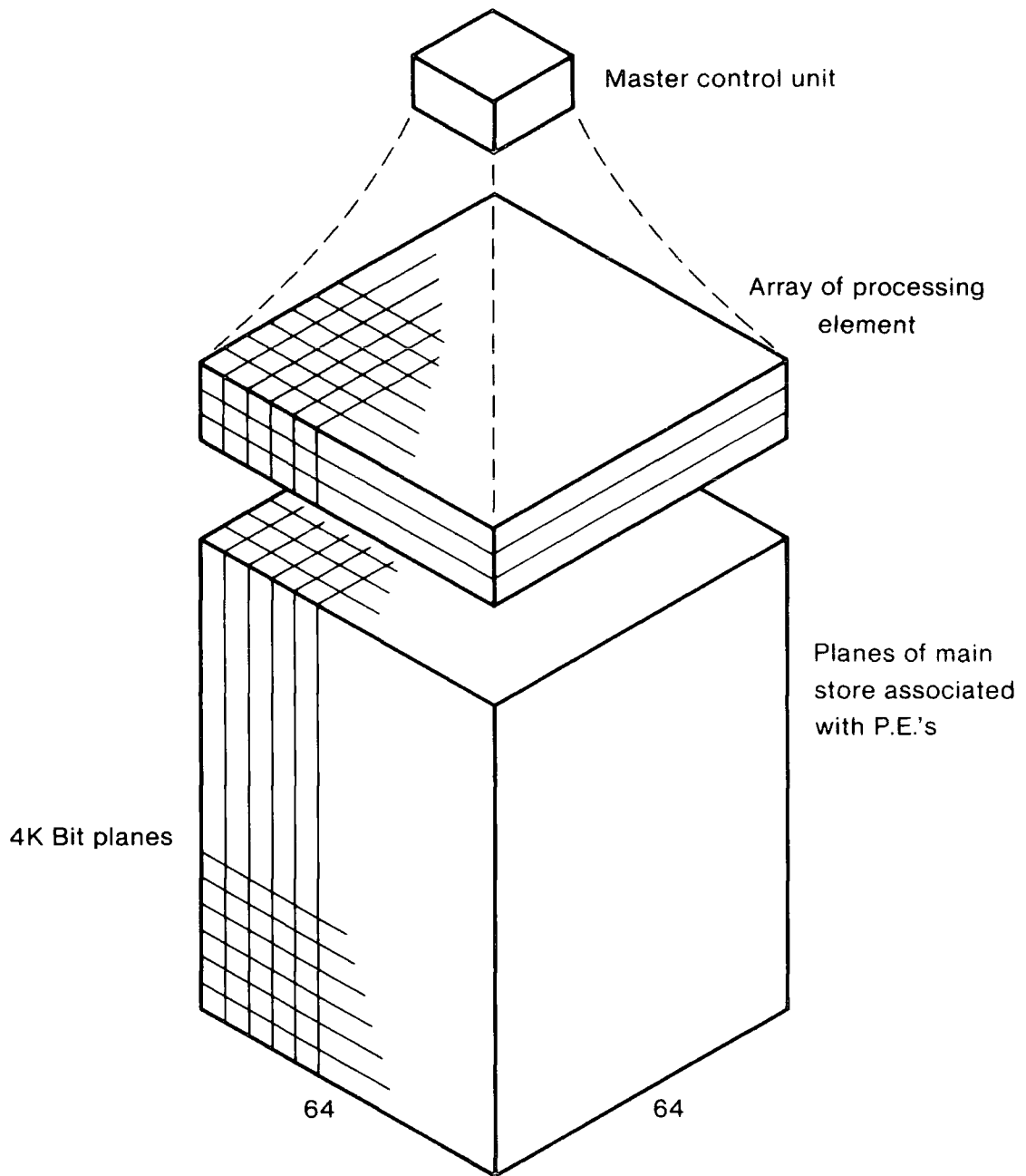
The pipeline approach described in 6.2.1 can be considered a processor of arrays; in contrast, the cellular strategy is truly an array of processors. Although ICL claim to be the only manufacturer with a commercially available product in this market - their Distributed Array Processor - the basic notion of an array of multi-operational processors which can execute the same instruction on many data items simultaneously has been known for many years. There have been several devices of this type built, most notably the Goodyear STARAN (Goodyear Aerospace 1974) and the Burrough's ILLIAC-IV (Thurber and Wald 1975). The ICL Distributed Array Processor (DAP) is similar in nature to these machines although there are two important differences:

1. there are 4096 processing elements in the DAP as opposed to 256 and 64 for the STARAN and ILLIAC-IV respectively;
2. the processing elements (PEs) are one-bit processors and

hence are limited in their applications - all data manipulation, other than single-bit logical instructions, must be performed by software.

Furthermore, the STARAN, which has been built under contract to NASA, is more flexible in its host-computing requirements. Whereas the DAP relies on the host machine being a powerful ICL 2900 series mainframe supporting the ICL VME/B operating system, the STARAN has been designed to connect to a standard mini-computer, typically the DEC PDP-11 series.

A simplification of the DAP architecture is shown in figure 6.2. The basic configuration comprises a 64 x 64 two-dimensional grid of PEs each with 4096 bits of associated local memory. The fact that there are 4096 bits of local memory and 4096 PEs is coincidental; future versions of the DAP will utilise large scale integration technology and increase the present two-megabyte capacities (64 x 64 x 4096 bits) up to eight-megabytes (64 x 64 x 16K bits). Each processing element is, to all intents and purposes, a central processor in its own right although the actual functions that can be performed are limited in extent. These machines have been christened as single-instruction-multiple-data stream (simd) computers; a single instruction only can be obeyed



SIMPLIFIED ELEMENTS OF THE ICL DISTRIBUTED ARRAY PROCESSOR

Figure 6.2

simultaneously in the multiple processing elements. Thus, two PEs cannot perform different tasks at the same time although they can be instructed to switch off for a specified number of machine cycles. The repertoire of the PEs is two fold: they can perform one-bit addition and a one-bit broadcast of data to one of their four neighbouring row and column PEs.

For purposes of computation, the DAP can be described as comprising:

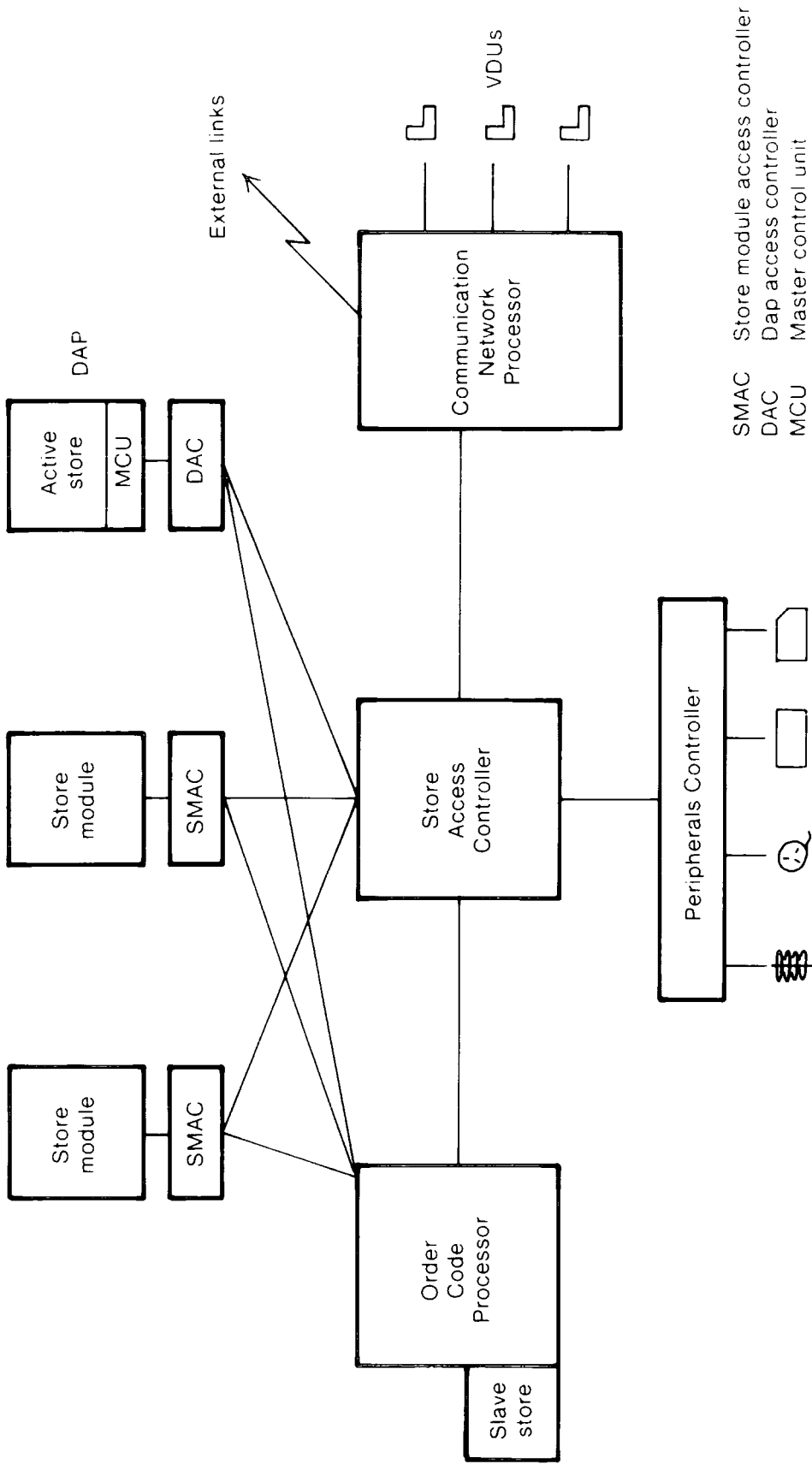
1. 4096 store planes of 64 x 64 bits;
2. the activity plane (A-plane) of 64 x 64 bits - this is one of the three planes comprising the PEs (see figure 6.2). The setting of a particular bit to '.TRUE.' in the A-plane instructs the associated PE to perform a given instruction; if the A-plane bit is unset (.FALSE.) then the PE remains idle and ignores any given instruction. The use of the activity plane has much importance in array-processing and, in particular, in geographic data processing; it acts as a 'mask' to determine whether each individual PE in the grid acts on an instruction or lies dormant;
3. the quotient and carry planes (Q and C-planes) of 64 x 64

bits - these two planes complete the three planes in each processing element. There are two basic operations that can be performed with these: the first is to add the contents of a store plane to the contents of the Q-plane (element by element) with the resulting sum being placed in both the Q-plane and the store plane with the carry bits set in the C-plane where necessary. The second operation involves shifting all the bits in the Q-plane one grid interval in the same direction. For example, the case shown below is a cyclic-shift north:

1	1	0		1	1	1	
1	1	1	---	→	0	1	0
0	1	0		1	1	0	

4. The master control unit (MCU) has the task of controlling the array of processors as a whole; it instructs them as to what they should be doing at any particular time.

The architecture of the DAP has been designed to fit neatly into an ICL 2900 series mainframe computing environment (a DAP will run with an ICL 2960 or larger). Figure 6.3 shows this relationship.



A TYPICAL ICL 2900 CONFIGURATION WITH DISTRIBUTED ARRAY PROCESSOR

Figure 6.3

To all intents and purposes, the virtual memory of the 2900 comprises a series of two-megabyte store modules, each with their own access controller to connect them to the order code processor (ICL terminology for central processor) and all other store modules. The DAP is considered by the host machine to be another two-megabyte store module except that it has the ability to perform its own specialised data processing; a DAP can be used as conventional 2900 store if array-processing capabilities are not required. To load the DAP with data is an identical operation to loading the 2900 main memory; the host mainframe must provide all input and output capabilities for the DAP, which is unable to perform any itself. In this way, ICL have been able to reduce the problems of the slow data loading associated with the pipeline approach (see 6.2.1).

Early releases of the VME/B operating system with DAP capabilities have necessitated a single-user batch-mode only for array-processing jobs. Recent modifications to the software have permitted a series of improvements:

1. the introduction of asynchronous input and output - hence I/O can be taking place whilst the DAP is still processing. The facility is achieved by cycle-stealing (i.e. the DAP will stop for each stolen cycle). At worst, this generates a twenty per cent degradation in

performance but on average is nearer to five per cent. Clearly, the ability to transfer eight-megabytes per second between the DAP module and other 2900 storage is advantageous to enable more efficient use of the DAP on tasks most suited to its capabilities.

2. The introduction of a Multi-programming environment - has given the ability to have two or more programs running in the DAP simultaneously. This facility operates by cycle-stealing as with (1) above.
3. The dynamic allocation of the DAP is a powerful recent modification - it is now possible to vary the amount of DAP store being required for a particular job as its demand varies. This is the first stage towards the multi-user interactive use of DAP facilities in real time.

6.2.2.1 Programming the DAP:

The DAP has been designed for efficient and easy implementation of user-written programs. Two programming languages are offered: APAL and DAP-FORTRAN. APAL is a low-level assembly language (ICL, 1979) providing for complete control of the hardware functions in a fashion

similar to the FPS-APAL mentioned in 6.2.1. DAP-FORTRAN is an extension of high-level standard ANSI FORTRAN. Almost all data processing can be readily programmed effectively in DAP-FORTRAN. The DAP hardware can execute all standard FORTRAN statements with the exception of formatted READ and WRITE operations; this must be left to the host ICL 2900 (ICL, 1981). There are a number of additional facilities in DAP-FORTRAN to provide efficient use of the array-processing capabilities. They evolve around the introduction of two extra variable modes: vector and matrix variables, together with an associated generalised indexing syntax to facilitate efficient use to be made of them. The distinction between the three modes can be declared as follows:

```
INTEGER SCALAR-INTEGERS, SCALAR-ARRAY(100), VECTOR()  
INTEGER SET-OF-VECTORS(,10), MATRIX(,)  
INTEGER SET-OF-MATRICES(,,15)  
(Similar declarations can be made for REAL, LOGICAL,  
and CHARACTER variables).
```

The dimensions of the array quantities are determined by the implicit architecture of the hardware. For the standard 64 x 64 DAP, each vector comprises sixty-four elements and each matrix comprises 4096 elements. Conventional arithmetic and logical operations are performed by software element by

element simultaneously. For example, if A, B and C are REAL matrices, then the syntax in DAP-FORTRAN to perform an element by element addition of B and C to produce the result in A will be:

```
A = B + C
```

In standard serial-code the above statement must involve a series of nested loops with associated indexing, thus:

```
DO 10 I=1,64
DO 10 J=1,64
10 A(J,I) = B(J,I) + C(J,I)
```

Mixed mode arithmetic and logical operations are quite legal in DAP-FORTRAN, providing there is no ambiguity in the result. For example, if A and B are REAL matrices and C is a REAL scalar, then:

```
A = B * C
```

means that A is the element-by-element product of matrix B by the scalar C.

The third significant feature of DAP-FORTRAN is masked (or logical) assignment; the assignment of one matrix to another can be masked with a LOGICAL matrix. For example, if A and B are CHARACTER matrices and MASK is a LOGICAL matrix, then the statement:

A (MASK) = B

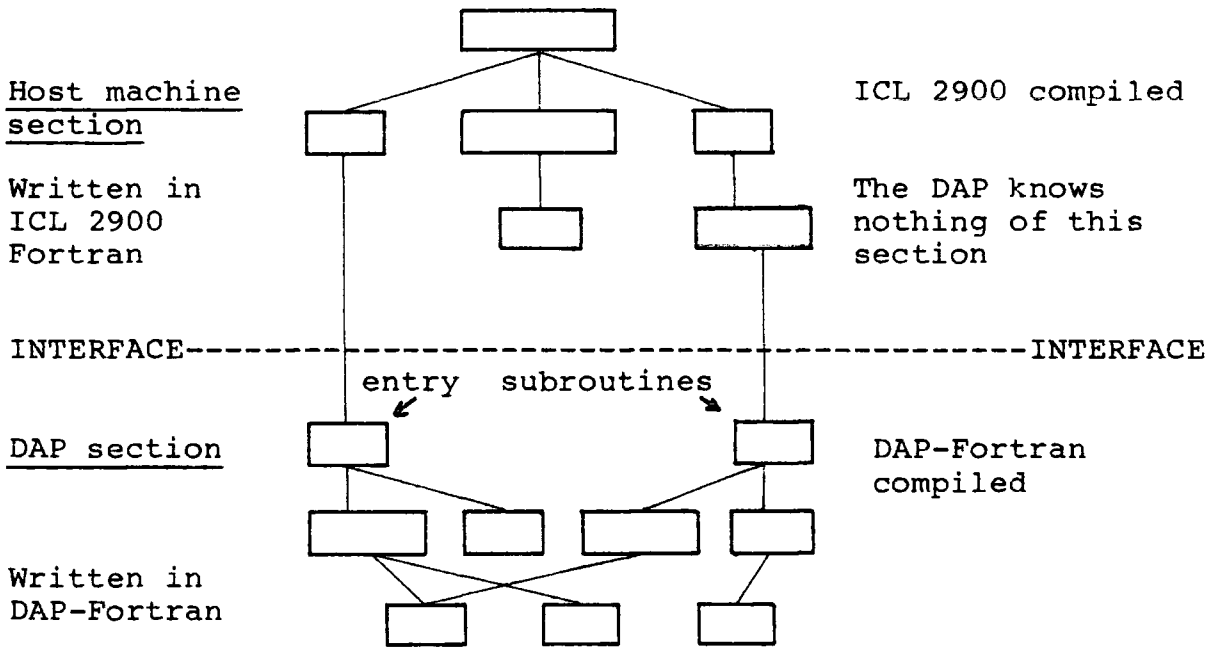
means that the element by element assignment takes place only where the elements of MASK are set; this technique uses the activity plane in the hardware described earlier.

Though it is not difficult to write programs to run on the DAP, a different approach to algorithm design is crucial for efficient use of the array processing capabilities (see section 6.4). Any programs in DAP-FORTRAN must fall into two parts:

- a DAP section for fast array-processing capabilities;
- a host section to handle I/O and sequentially-based tasks.

A modular structure of software is desirable with the use of a series of subroutines: entry into the DAP-sections are via ENTRY SUBROUTINES. Both halves of the program are compiled separately and are linked together before execution. A block-summary of DAP software design is given in figure 6.4.

Figure 6.4 Block diagram showing the modular nature and the separation between host- and DAP-related sections of typical DAP software



The compilation of the DAP section is made in three passes. The DAP-FORTRAN compiler converts the source code into APAL, the DAP assembler then assembles the APAL source into a semi-compiled format suitable for the consolidator (or link editor) to link the code to appropriate libraries producing an object module. This can then be linked to the host FORTRAN object module at run-time.

The system developed by ICL is versatile although the processing rates achievable are closely allied with data characteristics and task specifications. Providing the problem suits the hardware architecture then a cellular array-processor can provide many-fold increases in throughput. The author's experiences suggests that algorithm design is manifestly easier for certain tasks than with any sequential-based procedures. Conversely, if the problem characteristics and size are unsuitable, then performance degrades rapidly since any benefits of the parallel architecture are not being used to advantage. The amount of logic required to implement arrays of an inappropriate size makes for complex and costly software; it is difficult to implement completely generalised code to suit all user needs.

Nevertheless, experiments carried out during this research have proved that a DAP-like architecture has much potential in the image processing field. Certainly the handling of raster-based monochrome topographic data is well suited to a cellular array-processing environment. The appraisal tests have been carried-out on the first ICL Distributed Array Processor, installed at Queen Mary College in the University of London. This DAP is connected to a six-megabyte ICL 2980 mainframe which then supported the VME/B (version 6.21) operating system. The following sections report on the design of these tests and an assessment of the suitability of

array-processing techniques towards Ordnance Survey digital applications.

6.3 A RASTER-BASED DATA STRUCTURE APPROPRIATE TO THE ICL DAP

The tests carried-out on a conventional computer and described in chapter five have all been based on a raster buffer size of 5000 x 5000 elements. These dimensions do not fit neatly into the 64 x 64 architecture of the DAP. For convenience, a buffer of 4096 x 4096 pixels is more appropriate. In general, it is not desirable to amend data to suit hardware characteristics although array-sizes in multiples of DAP planes are easier to implement; each complete image can be stored within 4096 DAP planes (a DAP plane is a 64 x 64 'slice' of the DAP store module - see figure 6.2). However, in this instance, to reduce the raster dimensions to 16 million pixels does not alter image resolution unduly; for a 1:1250 scale map image, each pixel will represent 0.12 metres at ground scale and 0.10 mm on the map. All DAP-based research has been carried out with a 4096 x 4096 data set generated on the NUMAC IBM 370/168, using the serial-code described in 5.3.4.

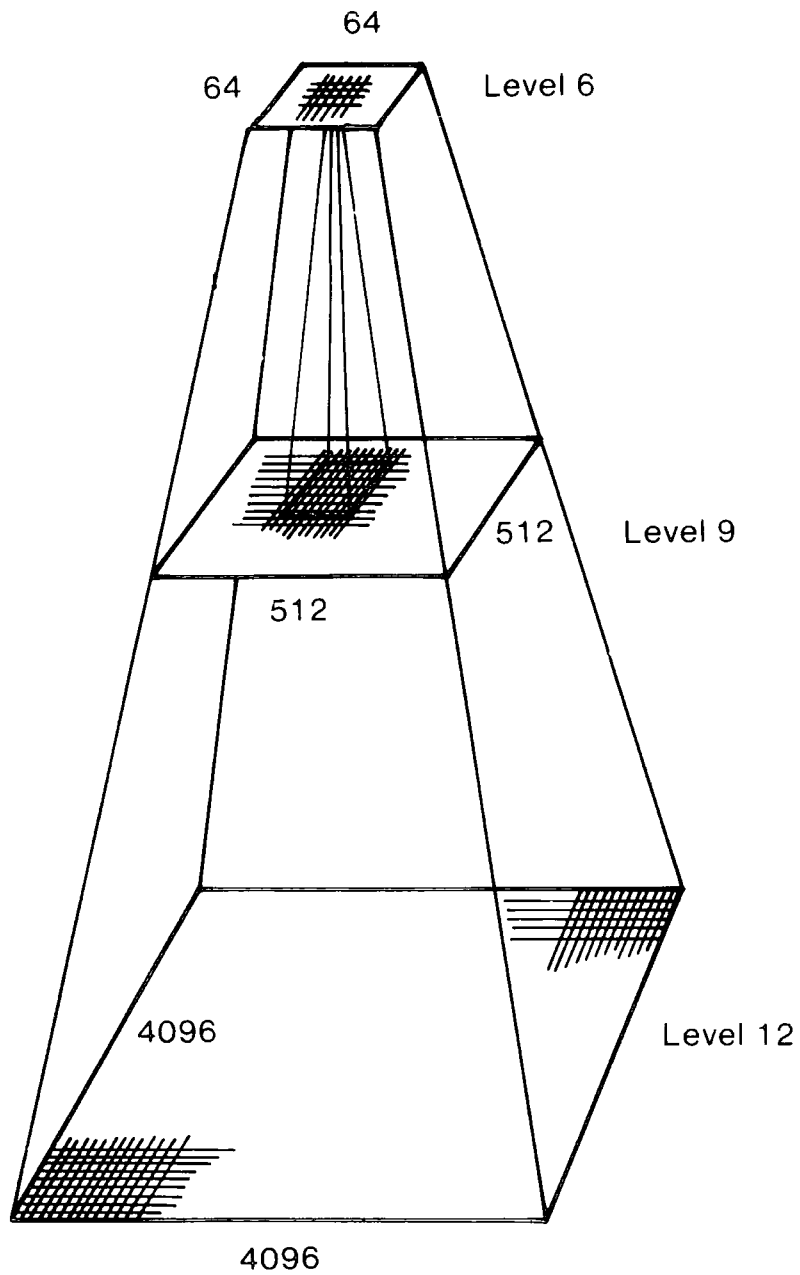
Clearly, existing data storage notations described in sections 5.3 and 5.4 are no longer appropriate to this

computing hardware: run-length encoding strategies are difficult to implement and eight-bits per pixel is too voluminous for such binary data - this would require sixteen-megabytes when only two-megabytes of DAP store are available. Because of this, it has been necessary to develop a new method of data structure to suit the architecture of the DAP.

The ability to store logical data (.TRUE. or .FALSE.) within a single binary digit on the DAP has enabled an efficient hierarchical data structure to be developed for Ordnance Survey raster-based map data. The individual raster buffers comprising the full DMB feature codes (see section 5.3) have been divided into a series of images - one for each feature-type - prior to writing these images onto tape. In this way, it is possible to store each image within a one-bit per pixel logical array. Yet, to hold a maximum of 109 raster buffers of 4096 x 4096 x 1 bit elements (one for each class of feature) is inappropriate with limited amounts of backing store. Fortunately each map sheet comprises only between twenty and thirty feature types (Adams 1979) and hence it is unnecessary to hold the raster files for codes which are not present. This strategy, when associated with the hierarchical properties to be described, has reduced data volumes by factors of ten-fold or more.

Figure 6.5

A HIERARCHICAL DATA STRUCTURE FOR RASTER DATA
ON A CELLULAR ARRAY PROCESSOR



The hierarchical data structure which has been developed is a logical extension of the quadtree ideas of Samat (1980) in which a total image is subdivided into tiers of information, where each lower tier is a subdivision of its parent. Ichikawa (1981) has attempted similar strategies for data compaction and rapid retrieval of digital images: preceding work, however, has been restricted to conventional computing hardware. It is appropriate to sub-divide the complete 4096 x 4096 raster map sheet into a series of levels (tiers) in which each comprises a collection of DAP planes. A description of the data structure is outlined in figure 6.5. At the most disaggregated level is the 4096 x 4096 image; this is level 12 ($2^{12} = 4096$). Level 6 is a 64 x 64 representation of the complete raster image ($2^6 = 64$) and fits exactly into one DAP plane. A level 6 representation of a complete 1:1250 map provides an index to the more detailed level 12 data; for each set pixel in the map-index, there will be an associated 64 x 64 DAP plane at level 12. In this way, the data structure developed is compatible with the DAP architecture since all the tiers are centred around an array-size of 64 x 64. Although this strategy has been primarily designed to suit the DAP, it has provided a minor form of data compaction: table 6.6 shows a summary of the rates of compaction for two Ordnance Survey 1:1250 test sheets - for comparison purposes one of the sheets (NZ 2642NE) was used in the serial-compaction tests described

in 5.4.1 and table 5.8. The other sheet (SO 9283SE) has been used extensively in the DAP-based investigations.

Table 6.6: A summary of the storage associated with the DAP hierarchical data structure

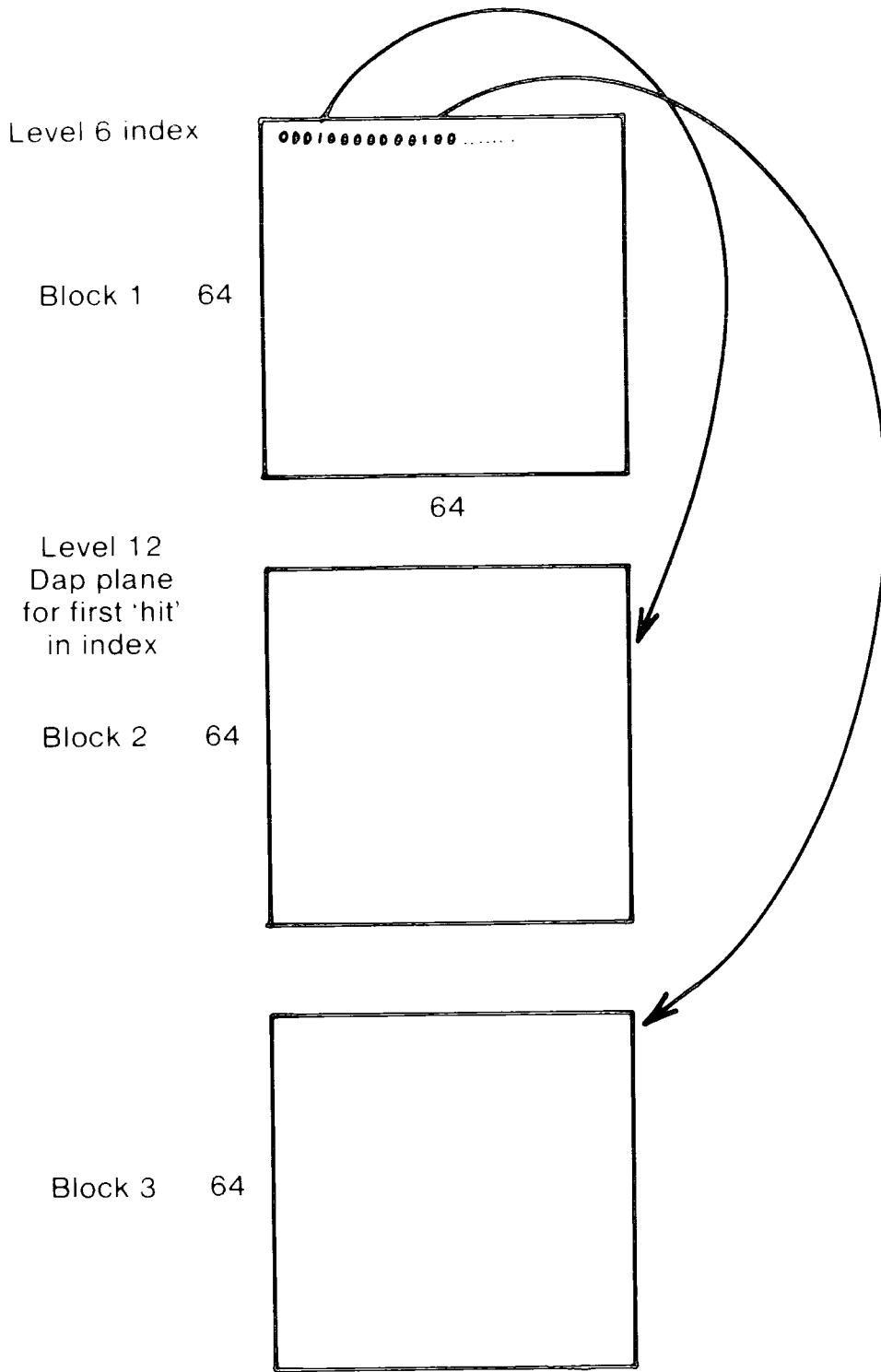
<u>-----</u>	<u>NZ 2642NE</u>	<u>SO 9283SE</u>
No. of DMB feature classes	30	21
No. of level 6 DAP planes	30	21
No. of level 12 DAP planes	6780	7004
Total storage requirements	3.4	3.6 megabytes
Times increase over run-length encoded data structure	4.4	4.7
Times increase over bit-map	1.3	1.5

6.3.1 The transfer of data to the ICL 2980

The hardwire communication links existing in 1981 between the two computing sites at Durham and London were inappropriate for the mass transfer of data. Access is available in batch-mode via the Metronet service and interactive connection is possible by telephone or the Rutherford Laboratories PACX system. The use of magnetic

tape was adopted for the transfer of the raster data between the two sites. The problems associated with data interchange on magnetic tapes have been pointed out by Visvalingam et al (1976). The transfer of the raster files as EBCDIC characters, grouped into appropriate data blocks, minimised these problems: both machines handle EBCDIC by default and both can read and write standard labelled IBM tapes. The hierarchical data structure described earlier was generated on the IBM 370 at Durham and written directly on to magnetic tape. With fixed length tape-blocks of 4096 characters comprising logical records of 64 characters, each DAP plane fits neatly into a tape block. The raster pixels are represented by EBCDIC characters ('1' for .TRUE., '0' for .FALSE.) and each of the feature-types were grouped into files separated by tape-marks. Each file comprised a level 6 (DAP plane) index for the whole map sheet, followed by the relevant level 12 data, which are also grouped into DAP plane units. A diagrammatic explanation of the tape format is shown in figure 6.7.

The size of each file is thus dependant on the number of set pixels in the index, since for every set pixel there will be an associated level 12 DAP plane. For example, if feature code 52 has 33 set pixels in the level 6 index, then the total file will comprise thirty-four blocks (2176 records): one block for the index and thirty-three blocks for each



MAGNETIC TAPE FORMAT SUITABLE FOR DAP ENTRY

Figure 6.7

associated level 12 DAP plane. The access and retrieval of each required DAP plane is trivial with this method of tape-formatting.

A serial-based FORTRAN program "QMCFORMAT" was developed to run at Durham which created the Queen Mary College transfer tapes. The source code for this is given in Annex 6a. The software takes as input a level 6 representation (64 x 64 pixels) and a level 12 representation (4096 x 4096 pixels) of each 1:1250 map sheet. It scans the level 6 index of each feature code and searches for set pixels. For each 'hit', an associated 64 x 64 partition of the level 12 data is written to the output tape. It was found necessary to develop appropriate mapping functions to determine which 64 x 64 level 12 partition is required for each level 6 pixel. A description of these mapping functions is given in Appendix 6.1. Standard labelled IBM tape formats enable files to be assigned names; each feature code represented on the map sheets is named 'FCn', where n is the DMB feature code integer value. In this way, it is simple to distinguish feature types on the ICL 2980 at QMC.

6.3.2 Data reformatting on the DAP

It is inappropriate to store the raster-based data as EBCDIC characters permanently; this is wasteful of storage space and was only a temporary measure to facilitate easy transfer of data to the DAP site. However, since the transfer tape is organised in units of DAP planes on a feature-by-feature basis, each DAP plane can be reformatted from one-EBCDIC character per pixel to an equivalent single binary digit. This notation is a facsimile of a DAP-FORTRAN LOGICAL matrix comprising a single bit plane of 64 x 64 elements from the 4096 planes available in the active store (see figure 6.2). The hexadecimal equivalent of the EBCDIC character '1' is 'F1', likewise EBCDIC '0' is hex. 'F0' and hence data reorganisation to DAP form is easy; a data compaction and reformatting routine merely involves extracting the least significant bit from the EBCDIC data. Annex 6b lists the DAP-FORTRAN source code developed to perform this reformatting operation: program FILES is the host FORTRAN and subroutine PACK is the DAP-FORTRAN section to perform the bit-stripping. The software can handle up to 300 DAP planes at one time (in variable 'FILE') and these are compressed into the variable 'PACKFILE'. The actual bit-stripping algorithm is performed in lines 117 to 118 of subroutine PACK. The parallel nature of the algorithm ensures rapid conversion. Since the code merely involves

logical bit-comparisons with no complex arithmetic, extremely fast data-reformatting is possible. For example, reformatting the EBCDIC file 'FC002' (minor buildings) has taken 800 milliseconds on the DAP. There are 252 level 12 DAP planes comprising the file, which represents 252 cycles through the bit-stripping loop; a conventional serial-processor would necessitate in excess of one-million cycles through a similar loop - once for each pixel - and required forty-six seconds of cpu time on the IBM 370/168 with Assembler programming. The results from the software in annex 6b are a series of DAP bit-planes in an appropriate format for immediate entry into the array processor for further manipulation. These files, in their much compressed form, are held on disk for convenience or off-line for archival purposes.

6.4 RASTER-BASED MANIPULATION APPROPRIATE TO AN ARRAY-PROCESSOR

The need for serious development of post hoc manipulation of raster-scanned data has been pointed out in previous chapters and elsewhere (Loodts 1981, Peuquet 1981, Weber 1980). Many of the limitations encountered in existing known research have been brought about by the inappropriate architecture of conventional computers (see chapter five). Fortunately a DAP-like machine is potentially well-suited to

the manipulation of raster-digital images and could provide a more appropriate means of performing the post hoc processing necessary. Regrettably, the design of the parallel-algorithms are grossly dissimilar to any serial equivalents. Indeed, many of the techniques adopted will be extremely difficult to amend for a sequential-processing environment.

A number of facilities are seen as requiring crucial investigation to develop a flexible, versatile raster processing system. These are: efficient graphic output, the introduction of appropriate topology into the raster "spaghetti" and a reliable conversion of the pixel data to vector strings. Though several systems are beginning to offer solutions to these problems (see chapter seven), it is believed that the DAP can provide useful assistance in all of these facilities. In particular, the DAP is expected to provide solutions to the following raster-based data manipulation:

1. Data reformatting, scale changes and rotation.
2. The identification and retrieval of all polygons.
3. The vectorisation of the polygon boundaries.

4. The determining of a spatial feature code for each polygon (possibly with human assistance).

6.4.1 DAP-assisted raster graphics - an example of a parallel image processing algorithm

The storage of the raster map-data in structures appropriate to DAP-processing is a necessary pre-requisite to any further data manipulation. The hierarchical nature of the data structure has provided two advantages: the development of versatile graphics facilities and a control over the volume of data used in any further processing. Efficient raster programming requires an ability to transfer between data levels (resolutions) near instantaneously; a parallel algorithm for this task is believed to offer better response than any sequential-bound process. For example, the ability to transform the level 12 data (4096 x 4096) to a level 9 representation (512 x 512) facilitates easy display of complete raster 1:1250 images on a Sigma colour raster display device with an addressable screen area of 768 x 512 pixels.

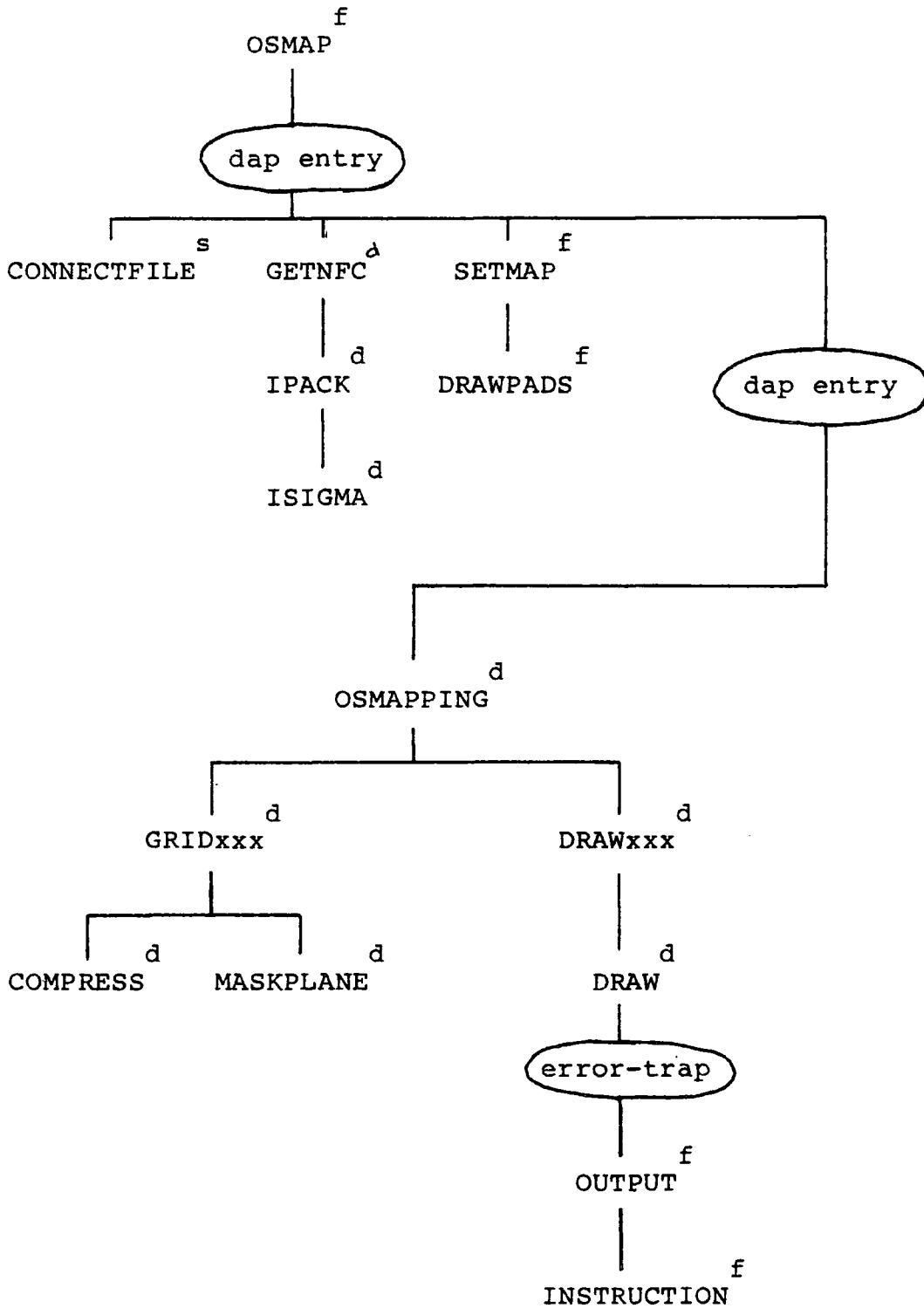
In essence a change of scale of raster data is obtained through a change in the resolution of the pixels. This is an

important difference to scaling in vector processing. Indeed, the ability to change the scale of raster data quickly suggests a basis for easy map generalisation of certain features such as buildings and roads. However, raster scaling is difficult; a scale change from a level 12 image to a level 9 representation on a serial-machine, for example, is a highly lengthy process. The algorithm involves sub-dividing the 4096 x 4096 image into a series of 8 x 8 "chunks" (512 x 512 of them), cycling through a pair of locally nested loops for each chunk to test the state of each pixel and assigning the equivalent elements in the resultant 512 x 512 image where necessary. In essence, the algorithm involves 16,777,216 cycles - once for each pixel; this has taken 36.9 cpu seconds on the IBM 370/168 at Durham using optimised Fortran code. Far more favourable results have been possible with parallel algorithms. The DAP has achieved the same end result in 75.4 milliseconds (a five-hundred times reduction) for the 'public-buildings' data which are limited in extent. Quoted DAP-times will be dependant on the number of DAP planes comprising each class of feature; even more extensive data (e.g. 'other buildings') have taken less than five seconds. Alternatively, the processing times on a serial machine will remain constant for both dense and sparse raster images. Annex 6c shows the DAP-FORTRAN listings of a suite of programs developed to perform this transformation between levels in the hierarchical data structure. With the

addition of relevant graphics routines, it is possible to display near instantaneously complete map sheets held as level 12 representations on a 512 x 512 colour display. The contents of each raster DAP plane are displayed as processing develops. Hitherto, any graphics displays of the raster data in this work has been limited to line-printer output (section 5.3.5) and Tektronix screen plots (section 4.2.3).

The modular design of the graphics software in annex 6c has facilitated easy interfaces with subsequent algorithms; much of the scale-changing software has been used extensively in further investigations. Figure 6.8 shows a block diagram of the modular graphics-suite with an indication of the array-processing sections.

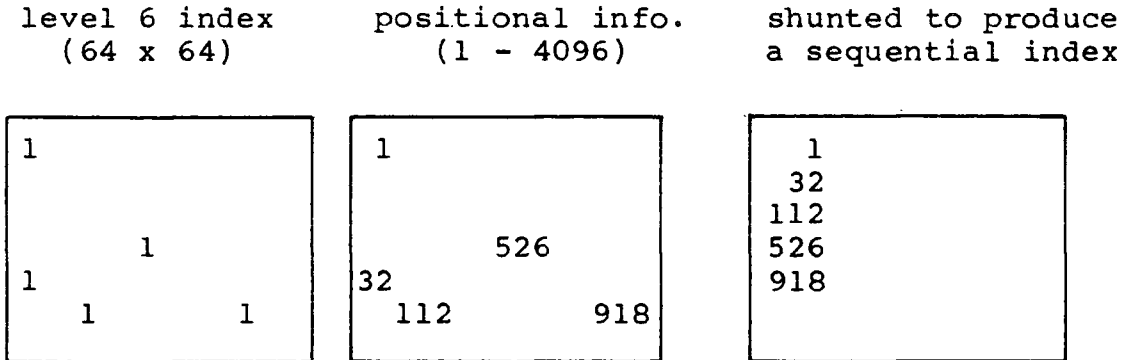
Figure 6.8: A block summary of the graphics routines developed for the DAP and listed in annex 6c



In figure 6.8, the suffices f = FORTRAN, d = DAP-FORTRAN and s = VME/B system control language.

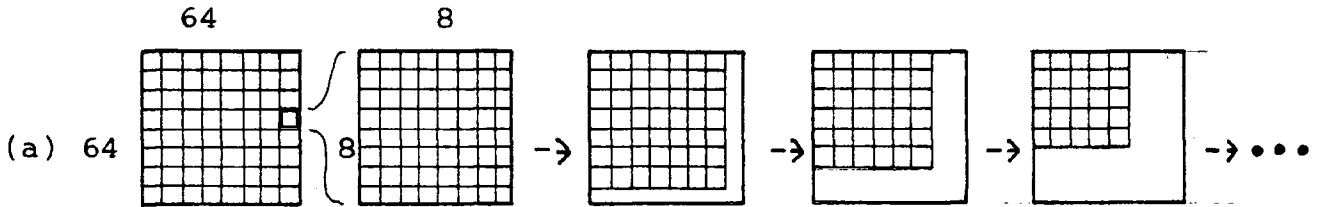
PROGRAM OSMAP is the ICL 2980 host program which controls the subsequent routines for the DAP and the graphics output. It requests which feature type is required and reads in the level 6 raster index for the complete map sheet. This is reorganised within the array-processor into a form more appropriate for rapid data indexing. This reorganisation is described below. Subroutine GETNFC takes the index and transposes each set-pixel into an equivalent positional address in an integer matrix (treating it as a long vector 1-4096). The positional data are then shunted (by function ISIGMA) to the upper-most part of the matrix providing a sequential index to the associated level 12 DAP planes. These stages are shown in figure 6.9.

Figure 6.9: The generation of an appropriate index to the level 12 data for an OS 1:1250 map sheet



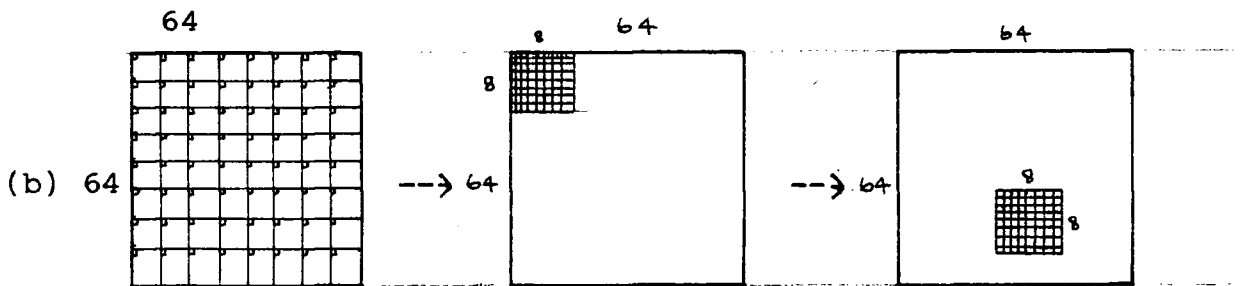
Each ascending element in the sequential index provides pointers to the location of each level 12 DAP plane; these can be read-in consecutively and restructured for the graphics display. Subroutine OSMAPPING is the DAP entry subroutine which hosts the collection of graphics modules. Three options have been included to transform the data from level 12 to levels 9, 10 and 11: GRID512, GRID1024 and GRID2048 perform these tasks respectively. The actual data reduction operation is performed in subroutines COMPRESS and MASKPLANE. COMPRESS compresses each 64 x 64 DAP plane into an 8 x 8 cluster of pixels (or 4 x 4 or 2 x 2 depending on which data level is required). Conceptually this is achieved in a folding operation over the whole DAP plane simultaneously: a graphical representation best explains this procedure - see figure 6.10.

Figure 6.10: A parallel algorithm to compress a 64 x 64 image to an 8 x 8



DAP plane sub-divided into 8 x 8 'chunks' of 8 x 8 pixels. 1-8 x 8 chunk of pixels folding operation by repetitive shifting and .OR. operations

For the whole DAP image, this produces:



Each significant bit representing the 8 x 8 chunk can be grouped together and positioned in the appropriate area of the (512 x 512) level 9 file and displayed on the graphics screen. Similarly, map sheet quadrants or octants can be displayed for the level 10 and level 11 data respectively.

Since the array processing hardware has no facilities to perform any data input and output, then control must be transferred to the host mainframe for the graphics display. An apt method of temporarily transferring this control, for each DAP plane, is to manually cause an error (or interrupt) in the DAP-FORTRAN at the point where I/O is required. This is achieved with the ICL 2900 extension to Fortran by using the 'ERROR' statement. Under normal circumstances an execution interrupt will cause the 2980 to interrupt the DAP, produce a diagnostic post-mortem report and halt the program. However, the VME/B executive provides facilities to trap interrupts and run alternative code; control is transferred to subroutine OUTPUT for the display of each DAP image, then returned to the DAP for the adjacent plane.

The detailed description of this algorithm serves as an example of the pertinence of DAP-based processing for binary-images. Such orders of processing-speeds are achieved by the elimination of costly arithmetic operations and only limited amounts of software looping. These stages are inherent in any serial-based equivalent code. Tests on both the DAP and the IBM 370/168 have shown a five-hundred times increase in performance with the parallel strategy; this gives a short delay in response to user requests which is essential for any interactive raster-system. An example of the results achieved on the SIGMA screen are given for levels

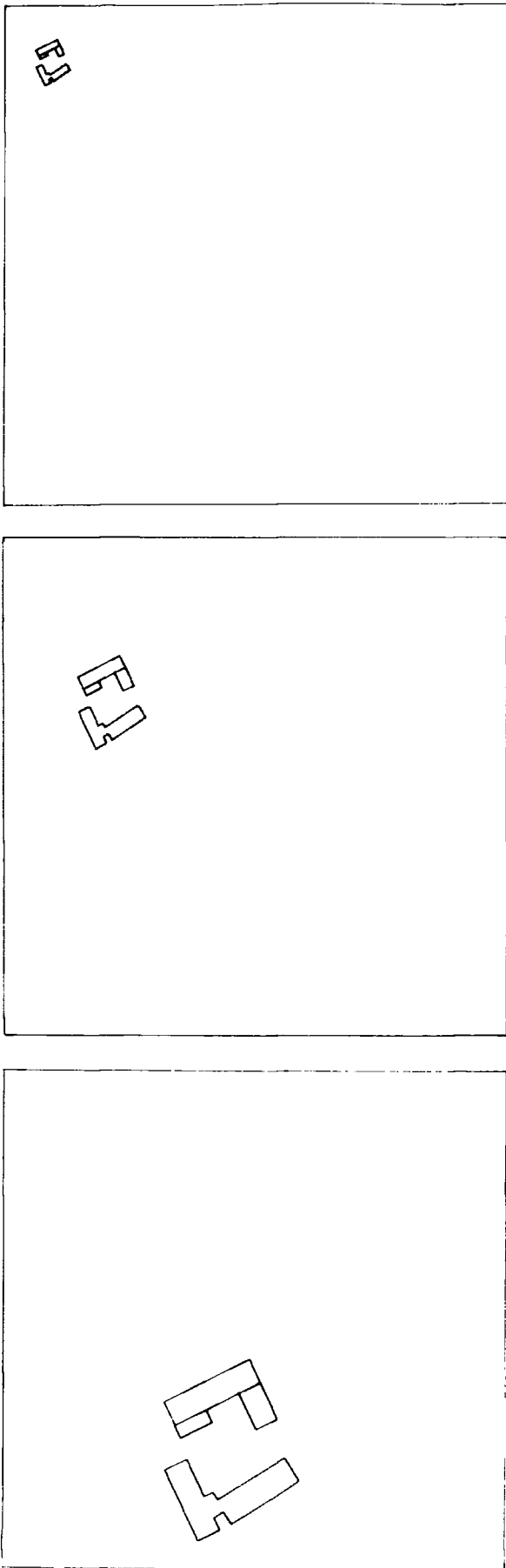


Figure 6.11: Raster display of two public buildings at three data levels:

- in order - (a) level 9 (512 x 512)
- (b) level 10 (1024 x 1024)
- (c) level 11 (2048 x 2048)

9, 10 and 11 data in figures 6.11a, b and c respectively.

6.4.2 The automated addition of topology to a raster image

The lack of any appropriate topology in scanned-images of the large scales monochrome O.S. data has been pointed-out in section 3.7. The design of appropriate parallel processing algorithms is expected to offer some solutions to this problem; software logic is arduous on a conventional computer (Rhind 1979) but far more viable on the DAP by virtue of the dimensional-connectivity of the hardware. In binary raster representations, each large scale map sheet can be treated as a series of adjacent regions or sub-parcels specified by a cluster of contiguous-pixels. Each sub-parcel has a valid description: a house, a footpath, a field and so on. This conception is difficult to implement on a serial-processor but fits neatly into the grid-of-processors of a DAP-like architecture. Furthermore, the ability to change between data-levels, as outlined in the previous section, has importance here in controlling the size of the image being held in the DAP thereby varying the scale of the image being processed. Clearly, an image with a low resolution level will have fewer DAP planes of 64 x 64 per map sheet than will a higher resolution level. Thus, it is possible to control

the number of sub-parcel units held in a single DAP plane at any one time by virtue of the scale-changing concepts described earlier.

Tests have been carried-out with parallel algorithms to digitally identify and isolate the sub-parcels quickly and hence to sub-divide the complex map images into their atomic raster-fragments. The boundaries of these fragments can be displayed on a raster colour display, vectorised to produce a file of polygons for other tasks or classified by shape, size and perimeter lengths in pixel form to facilitate a crude form of automatic feature coding. Theoretically, at least, it will become possible for users to request specific types of sub-parcel in a raster or vector format or they can be amalgamated to produce more appropriate spatial units such as property parcels as required by H.M. Land Registry. A description of the algorithms developed now follows; listings of the actual DAP-FORTRAN source code can be found in annex 6d and 6e.

6.4.2.1 Isolation of the sub-parcels:

The method of isolating each sub-parcel is best described for a single 64 x 64 sub-section (DAP plane) of the whole image; the theory can be expanded to the collection of DAP planes comprising a complete map by means of software

looping. Tests have been carried-out using data in levels 9, 10 and 11 for the sub-parcel identification process; a level 11 representation (2048 x 2048) has been found to be most appropriate, giving an adequate number of sub-parcels (c. fifty) per DAP plane. for easy handling of buffer storage, frequency of program interrupts, etc.

Each sub-section (64 x 64) of the map image fits into a LOGICAL matrix variable exactly: the pixels are unset for areas inside the sub-parcels and set where each boundary is located (this bit-map concept can be generated directly by most scanning devices). Each sub-parcel unit is identified individually by means of an expansion process within the sub-parcel until the complete boundary has been detected. The DAP plane directly beneath the one holding the binary map image is initially set to .FALSE. throughout. The first step is to find a pixel which lies inside the first sub-parcel unit; an in-built function in DAP-FORTRAN 'FRST' takes a logical matrix and returns a second logical matrix which is wholly set to .FALSE. except for the position of the first .TRUE. detected in the input matrix. Thus, if logical matrix 'MAP' stores the map image and the logical matrix 'SEARCH' is the DAP plane directly beneath MAP to be used when isolating the sub-parcels, then the following DAP-FORTRAN statement will identify the first pixel inside the first sub-parcel and place this in SEARCH:

```
SEARCH = FRST (.NOT. MAP)
```

With a parallel approach this single point is then repeatedly expanded to fill the first sub-parcel, stopping its propagation wherever it meets a boundary. The function 'NBH' in annex 6d performs this expansion; the process involves a shift north, south, east and west of the single point in SEARCH to set its nearest neighbours. This procedure is repeated in-parallel for the newly-formed five-pixel cluster to produce a thirteen-pixel cluster and so on. The code shown below gives a simplified example of the algorithm necessary:

```
LOGICAL MATRIX FUNCTION EXPAND (AREA)
LOGICAL AREA(,)
EXPAND = AREA .OR. SHNP(AREA) .OR. SHEP(AREA)
C      .OR. SHSP(AREA) .OR. SHWP(AREA)
RETURN
END
```

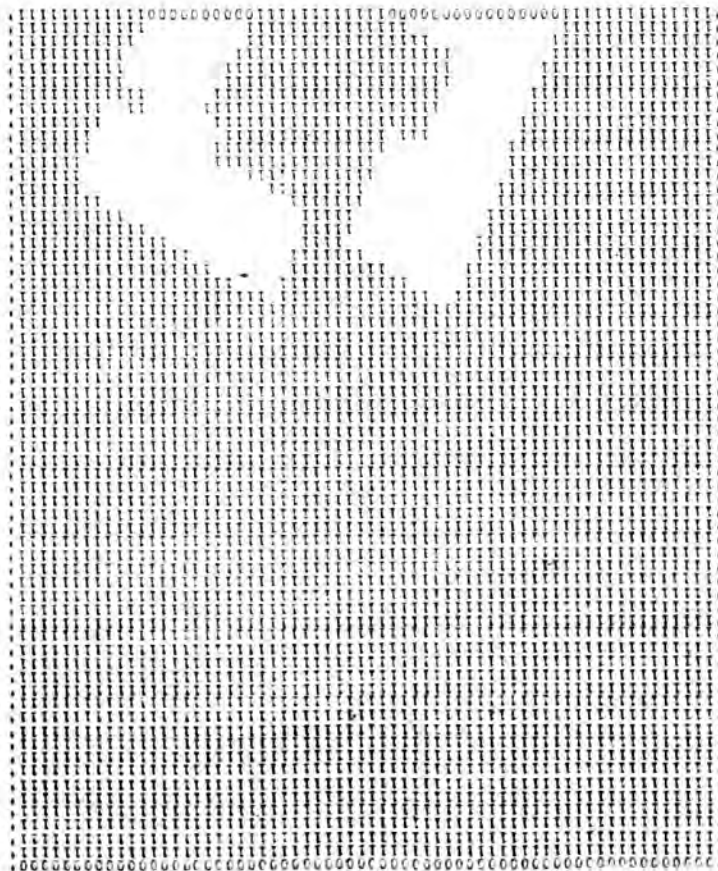
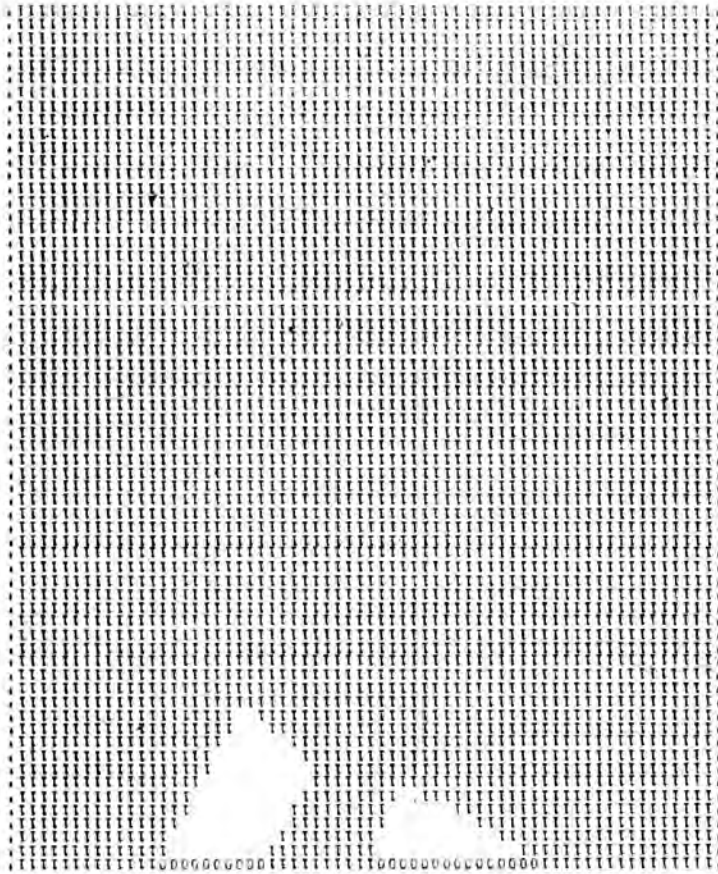
The functions SHNP, SHEP, SHSP and SHWP are in-built DAP-FORTRAN routines to shift a logical bit-pattern north, east, south and west respectively. The boundary information held in MAP provides a logical mask (see 6.2.2) to stop the propagation of the expanding pixel venturing outside the region of the sub-parcel. Thus, the complete code to find the first sub-parcel in MAP might be:

```
SEARCH = FRST (.NOT. MAP)
DO 10 I=1,64
10 SEARCH (.NOT. MAP) = EXPAND (SEARCH)
```

The mask '.NOT. MAP' in line 10 ensures that pixel expansion in SEARCH only occurs within the sub-parcel boundaries; clearly the expansion process will "spill-outside" the region if there is a one-pixel gap in any of the sub-parcel boundaries.

The same process is repeated for each sub-parcel in the map sheet; each is "painted-out" from the master file and moved to some appropriate sub-file as it is identified. In this way it is superficially simple to introduce some minor topology into the raster "spaghetti" quickly. The above process has been applied to a 4096 x 4096 raster representation containing 1677 sub-parcels and took three seconds of DAP time to isolate each into a series of sub-files. Figure 6.12 shows a line-printer representation of two DAP planes during the sub-parcel extraction process. In this example, the large-outer region of the map area (represented by '1' in the extract) was being identified.

Figure 6.12 Lineprinter representation of two DAP planes whilst extracting sub-parcels from the master raster image of map data.



6.4.2.2 Conversion of raster sub-parcels to vector polygons:

Once this limited amount of structure has been introduced into the binary raster map images, post hoc vectorising or feature coding is straight-forward. It is trivial to vectorise the boundaries of the raster sub-parcels on the DAP in a simple, logical and useful form. Harris et al (1981) describe a serial-based conversion system which produces a segment and node format although most other commercial solutions produce poorly structured, ad hoc line-work (see results described in section 7.4.2).

Vectorising developments by the author have taken on two distinct courses: the selection of the appropriate method is dependant on the types of use to which the vector files will be put. Each method involves initially defining the sub-parcel boundaries in raster-form; the series of sub-files comprising the whole map image are representations of the internal areas of each sub-parcel. The boundary location can be established by means of the point expansion idea described for the sub-parcel identification process. The function EXPAND (see earlier) is applied to each sub-parcel unit to identify the boundary pixels extremely quickly (c. 600 nanoseconds per DAP-plane).

The simplest method to convert the boundary pixel data into x,y coordinate pairs is achieved when linear-connectivity is unimportant; the resultant vector file comprises a collection of disordered coordinates held scan-line by scan-line. This vector-format is quite acceptable for efficient storage of the raster sub-parcels and is, indeed, near ideal for rapid display on a raster display device; the resultant image appears scan-line by scan-line for the whole map sheet in units of 64 x 64 DAP planes. The DAP-FORTRAN to perform this task is trivial. Two integer matrices are declared to provide the base-coordinate system, thus:

$$X = \begin{bmatrix} 0 & 1 & 2 & 3 & . & . & . & . & 63 \\ 0 & 1 & 2 & 3 & . & . & . & . & 63 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & 1 & 2 & 3 & . & . & . & . & 63 \end{bmatrix}$$

$$Y = \begin{bmatrix} 63 & 63 & 63 & 63 & . & . & . & . & 63 \\ 62 & 62 & 62 & 62 & . & . & . & . & 62 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 1 & 1 & 1 & 1 & . & . & . & . & 1 \\ 0 & 0 & 0 & 0 & . & . & . & . & 0 \end{bmatrix}$$

The generation of the x,y boundary coordinates is straightforward; the boundary DAP planes are treated as logical masks for the coordinate assignment, thus:

```
INTEGER X(,), Y(,), XCOORD(,), YCOORD(,)
LOGICAL BOUNDARY(,)
XCOORD=0
YCOORD=0
XCOORD (BOUNDARY) = X
YCOORD (BOUNDARY) = Y
```

In this way the resultant matrices XCOORD and YCOORD will comprise the disordered x,y boundary coordinates respectively.

Clearly this method of raster to vector conversion is unsuitable for most post-manipulation. Many vector-systems (e.g. see Baxter 1980, Waugh 1980) expect a series of polygons as input data; the use of vector-based pen-plotting devices require the linear connectivity inherent in vector strings to work efficiently. Tests have been carried out on the DAP to generate a series of linearly connected polygon files from the raster sub-parcel data. It is less easy to utilise parallel-processing techniques for this process; the task requires a line-following algorithm which, in essence, is sequential in nature. A listing of the relevant DAP-FORTRAN code is at annex 6e; subroutine GINOV has been written to chain the boundary pixels into a series of vector polygons. The algorithm works in a similar manner to the

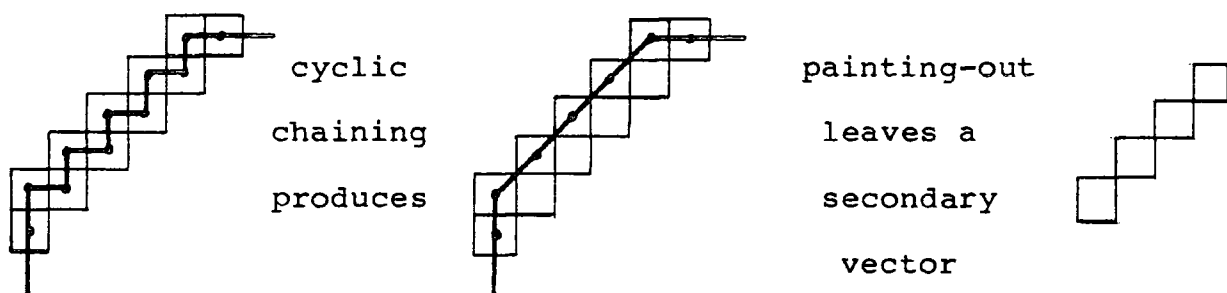
point expansion process. For each logical DAP plane comprising a sub-parcel boundary, the first (FRST - see earlier) point is detected and the associated x,y coordinates are written into a vector-buffer file. A cyclic search in a clockwise rotation is performed to detect the next pixel in the vector chain. As each new pixel is detected, its x,y coordinates are transferred to the vector-buffer and the boundary pixel in the raster image is "painted-out". This procedure continues until the start position is reached; this will always occur since the resultant vector will be a closed polygon by definition of the sub-parcel units.

A number of specific features in the algorithm are important; these are described below.

1. Each vector polygon is not restricted to one 64 x 64 DAP plane; it frequently overlaps into adjacent regions. The cyclic searching operation has been introduced inside a moveable window. Each time the vector coordinates reach the edge of the DAP plane, the search window moves into the next half-DAP plane for continued searching. In this way it has become easy to chain very long vectors (e.g. roads) across many DAP planes; the work has been carried-out on level 11 representations (2048 x 2048) which comprise 32 x 32 logical DAP planes for the complete 1:1250 sheet. The windowing strategy is coded

in subroutines WINDOW, WINDOW-COORDS and STORE-WINDOW (see annex 6e).

2. Implicit in normal, pre-skeletonised raster representations of boundary data are lines which are greater than one-pixel in width. As the line-following operation is taking place, with its associated "painting-out" facility, it becomes necessary to check that adjacent pixels are also removed if the need arises. An example of the problem is shown below:



A secondary logical matrix 'ALLPTS' is necessary to keep a tally on the route taken by the line-following process; as soon as the polygon has closed, all the remaining secondary vectors are deleted to enable vectorising of the next sub-parcel to begin.

The resultant file contains vector polygons of a higher quality by these methods than by more traditional raster-to-vector conversion methods since, by first isolating

the sub-parcels in raster form, ensures that the conversion software need not be concerned with line-junctions (nodes) or the hindering effects of adjacent polygons. Once generated, the linear features can be interfaced to existing serial-based software such as GIMMS (Waugh 1980) or plotted directly on an appropriate display/hard-copy device by means of valid graphics software: GINO, *GHOST and PLOTSYS are examples. The resulting data structure will comprise duplicated segments and (exceptionally) polygons which, according to Arms (1970), can be broken down into more appropriate vector-segments easily.

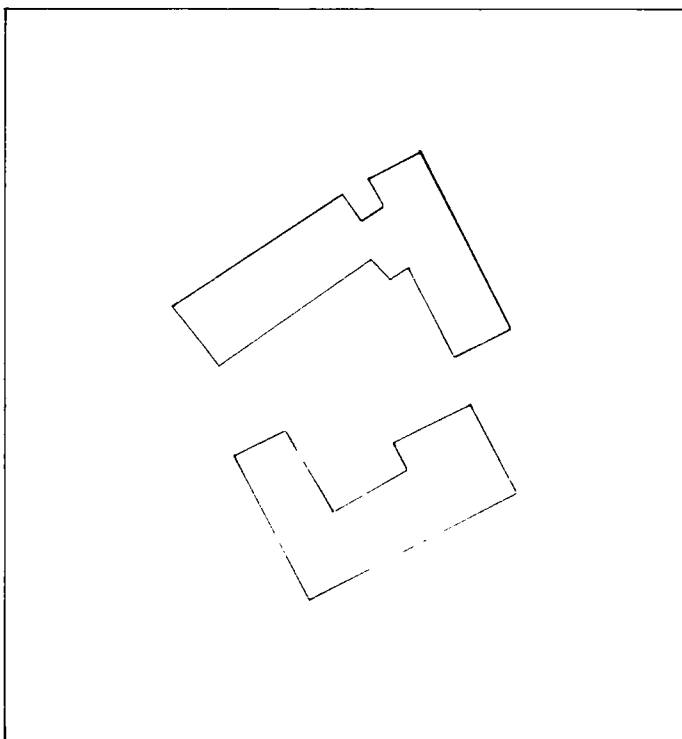
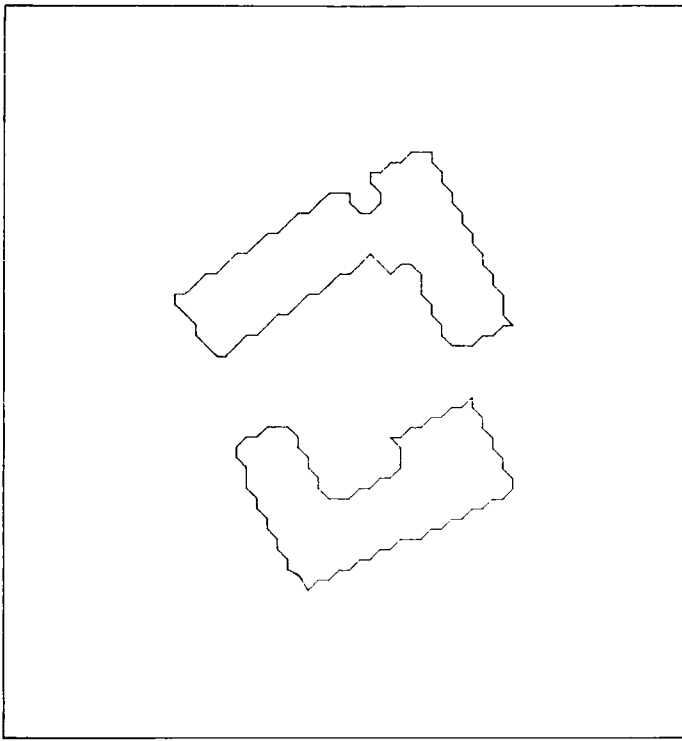
Additionally, post hoc filtering is desirable; the vector data emanating from the DAP routines characteristically comprise too many x,y points for efficient storage and plotting. There will be a coordinate pair for each pixel in the raster image - the majority of these points can be filtered to leave only prominent feature end-points, corner points or those necessary to accurately define curves. There have been several studies into efficient point-filtering (see Douglas and Peucker 1973, Opheim 1980); tests have been carried-out by this author using serial-software written by Douglas when in the University of Ottawa. The vector files generated by the DAP were transferred to the IBM 370 at NUMAC through the Metronet communication link and filtered to produce data suitable for a pen-plotter. Figure 6.13 shows

Figure 6.13 Raster-to-vector

conversion of sub-parcel boundaries.

In order (a) by DAP (before filtering)

(b) after filtering.



the results of the vectorising process (a) from the DAP and (b) the equivalent post-filtered version for two prominent public-buildings on an Ordnance Survey 1:1250 map sheet. Such post hoc filtering is a process common to all raster-to-vector conversion software (see Harris et al 1981, Peuquet 1981); it seems likely that the inherent linear natures of these features will result in such generalisation being most suited to a serial-process. Research has examined the potential of DAP-based software to speed-up this process with little success at present. In any event, the existing serial-code is not unduly lengthy, taking fifty cpu seconds on the 370/168 for a complete Ordnance Survey 1:1250 plan.

6.4.2.3 DAP-assisted sub-parcel feature coding:

The vector data emanating from the procedures described above are appropriate for immediate entry into a vector-based interactive edit-station. Clearly, any feature-coding (i.e. the assignment of attribute codes to features such as line segments) will still require operator intervention. Existing results achieved on the DAP suggest that appropriate parallel-algorithms can assist in this arduous post hoc stage with the raster sub-parcels data. It is almost certainly impossible to detect and identify all members of the complex scheme of 109 feature codes currently adopted by the Ordnance Survey, since some of these are only identifiable through

recourse to adjacent text or to general context. There are, however, some advantages in categorising the sub-parcels into simple groupings such as roads, buildings, railways, water-courses, fields and so on. This, at least, will be a significant start towards the development of a computer-aided feature coding system for easier validation by an operator afterwards: it will also permit the computerised partitioning of scanned map data into smaller and more manageable files.

The automatic categorisation of the sub-parcels involves an analysis of the size, shape and perimeter lengths in pixel form. To obtain size and perimeter lengths in raster is trivial on the DAP; it merely involves a count of the pixels involved and there is an appropriate function in DAP-FORTRAN to achieve this. The determination of parameters describing the shapes is not unduly difficult; consideration is given in most works on quantitative geography techniques - see in particular, Haggett et al (1977 pp 309-312) and Taylor (1977 pp 41-44). In the image analysis area of remote-sensing data (see Rosenfeld and Kak 1976 section 8.3), attempts are made to match pixel-patterns with similar structures held in a pattern library. Similar procedures were investigated with the sub-parcel data on the DAP. A series of pixel patterns comprising two, three, four or more pixels of many shapes and orientations were held in a pattern library. In turn, a count was determined of how many of each pattern-type will

fit neatly into the sub-parcel areas thereby providing a type of spatial signature for each one. It is possible to match-up the attained signature with a predetermined list to classify the sub-parcel type. The orientation of the sub-parcels can be expected to have prominent effects on the derived shape parameters; different arrangements of pixel boundaries will give very different shapes. However, the success is allied with the number of pixel-patterns adopted in the comparisons. All derived parameters must be ratio values of the pattern-count and the area of the sub-parcels. Providing pixel-patterns in all orientations are included in the algorithm then sub-parcel classification is manifestly apparent. Clearly a road parcel differs considerably from a building parcel; a sixty to eighty per cent success rate in such circumstances is hoped for, although at present any development has been necessarily restricted to artificial data generated from existing vector files. The strategy adopted incorporates the Sigma colour raster display device to show all sub-parcels of a specific variety. A human-operator can then accept or reject the DAP's categorising interactively. The resultant 'spectral signatures' are displayed on the SIGMA graphics screen for each uniquely referenced sub-parcel unit (see figure 6.14).

The parallel algorithm to perform the auto-feature-coding involves disaggregating each sub-parcel into its constituent

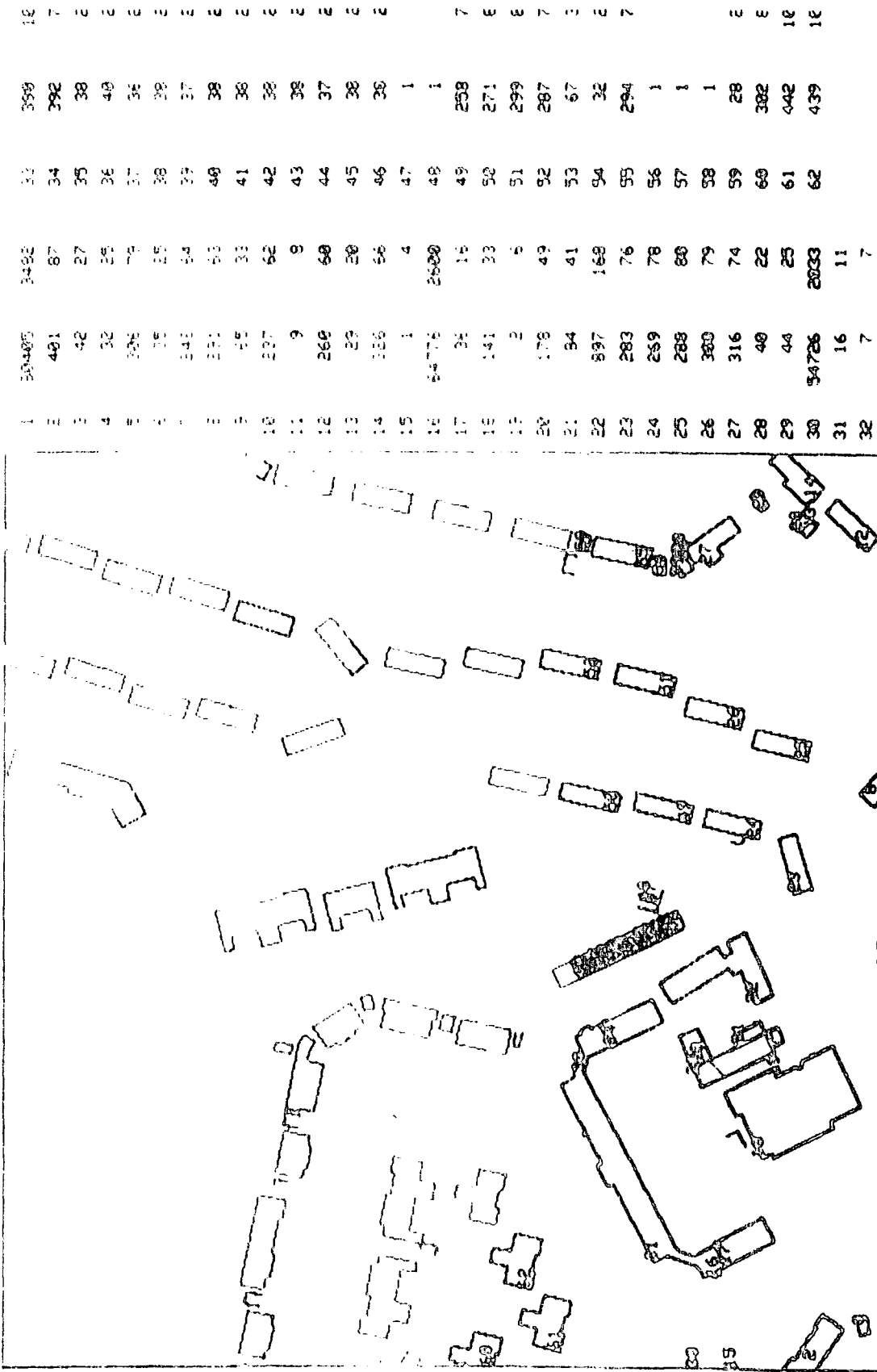
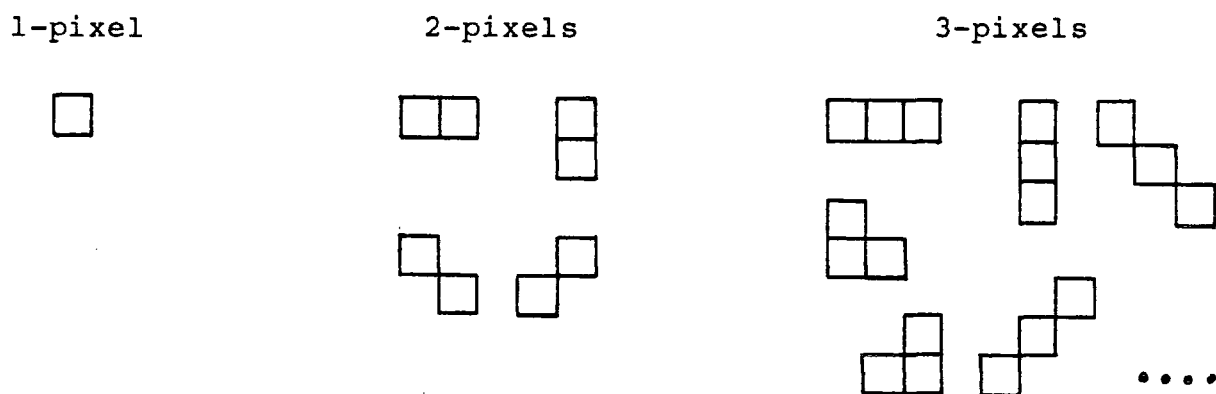


Figure 6.14 Example of interactive display of Sigma raster screen showing sub-parcel references with associated spectral-type signatures.

pixel formations. Some of the pixel clusters being examined are given below in figure 6.15.

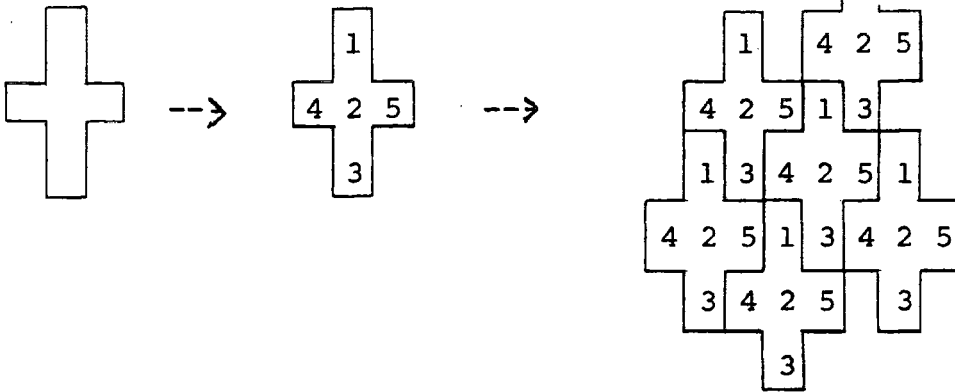
Figure 6.15: Types of pixel pattern to measure shape



A measure of the sub-parcel shape is determined by counting how many of these basic pixel-clusters will fit completely inside a sub-parcel area. Clearly, elongated pixels (e.g. roads) will have a higher frequency of the 3-pixel elongated patterns than building parcels. Similarly, the buildings comprise a much greater proportion of right-angles than do roads, fields, rivers, etc.

The implementation of such a task is well-suited to the pixel-connectivity associated with the DAP's architecture. Each chosen pixel-pattern must interlock neatly to ensure a consistent coverage over the whole DAP plane. Current tests have involved assigning a unique reference to each pixel in the pattern-cluster. A simple example is shown in figure 6.16.

Figure 6.16: The interconnection of pixel-clusters to determine the shape of sub-parcel units



-----> scan lines

It can be seen in figure 6.16 that for each raster scan-line, the series of interconnected pattern-clusters generate a numeric series: in this example it is - 1, 3, 4, 2, 5; 1, 3, 4, 2, 5;... Thus, the algorithm necessary to describe each pattern requires two parameters:

1. number of pixels in the cluster-pattern (a scalar variable);
2. the x-shifts necessary to place them in the correct positions relative to some local datum (a vector variable).

The scalar variable representing the previous example is 5 and the vector-shifts are 0, 2, 4, 1, 3; this is shown graphically below:

```
1 ----> shift 0  )
2 ----> shift 2  )
3 ----> shift 4  )- 1 4 2 5 3, 1 4 2 5 3, ...
4 ----> shift 1  )
5 ----> shift 3  )
1
2  etc...
```

Since the pattern-cluster comprises five pixels, it is held within five LOGICAL (1-bit) DAP planes which can be tested with an inclusive 'OR' to give the complete pattern; each unique pixel reference in the series is stored as layers within its unique DAP plane.

The frequency of patterns falling completely inside a sub-parcel unit involves a count of the patterns which remain when 'masked' by the DAP plane comprising the sub-parcel image. The DAP-FORTRAN to perform this task is trivial:

```
INTEGER FREQUENCY
LOGICAL SUBPARCEL(,), PATTERN5(,)
FREQUENCY = SUM (PATTERN5 (SUBPARCEL))
```

The DAP-FORTRAN function SUM gives a count of the set pixels in PATTERN5 (which is the right-most edge of the cluster pattern) for all areas where SUBPARCEL is set. This value, when allied with the results achieved from other

cluster-shapes, can categorise the sub-parcel image to some (presently unknown) extent. Initial investigations have involved categorising known sub-parcel units by means of the OS DMB data converted to raster mode (see section 5.3). The theories are easily transferable to data which have come from high-quality raster scanners provided that no gaps exist in the raster line-work. For this reason, an obvious next step is to classify sub-parcel regions within data which have emanated from an expensive and sophisticated raster scanner.

6.5 CONCLUSIONS

The hardware architecture described in this chapter is novel. The benefits gained in complex numerical applications are not completely assessed at the present time; certainly there is evidence (Klette 1979, Marks 1980, Reeves 1980) to suggest much potential in image-processing work. These techniques are in their development stages though early results in this research have been very promising for raster-cartography. Since the DAP is performing the same operation simultaneously, then extremely fast throughput-rates have been possible. It is difficult to provide comparisons with serial-based equivalents (see Parkinson 1978) since the majority of algorithms being used are not amenable to the architecture of a conventional

computer but tests carried out on an IBM 370/168 have been generally orders of magnitude slower.

On the whole, any conclusions reached on the characteristics of these new-generation machines must include a mention of their versatility and relative ease of use. Extremely powerful operations can be coded in a few lines of DAP-FORTRAN which are four or five times shorter than any serial-code. The result is short, easy to understand software. It must be pointed-out that generally any programming tasks require much different algorithm logic; it is easier to re-design the software from first principles than to attempt to convert existing serial-code for a parallel computer.

Most of these tests were based on 'clean' data although their characteristics are directly transferrable to equivalent data emanating from a raster-scanner. Tests carried-out on high-quality scanned data are currently in progress with the view to similar results being attained. On the basis of results to date, it seems not unreasonable to expect that a parallel-based raster system will be the norm by the late eighties.

Many of these early developments are highly machine-dependant on a device which is currently expensive as

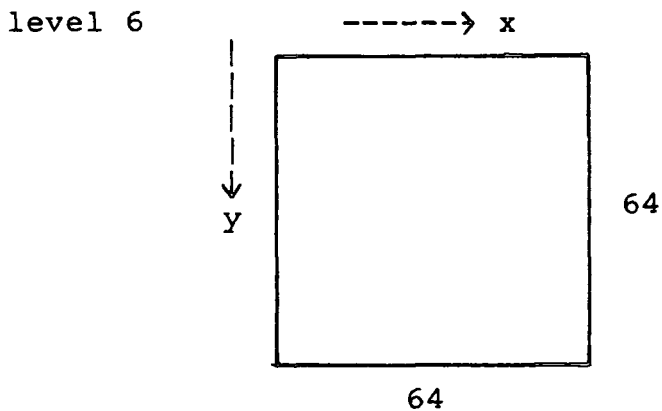
a result of only a limited number being sold at the present time; a DAP costs c.600,000 pounds sterling + associated costs for an ICL 2900 host mainframe. As has been pointed out in section 6.2.2, the DAP is the first of its kind, though the Goodyear Staran used in The US Engineer Topographic Laboratories has certain features in common. Others are bound to follow before too long; already Braunleder and Kober (1981) describe a similar machine based on an array of micro-processors. The design of parallel algorithms must be made now in preparation for these new-generation machines.

APPENDIX 6.1

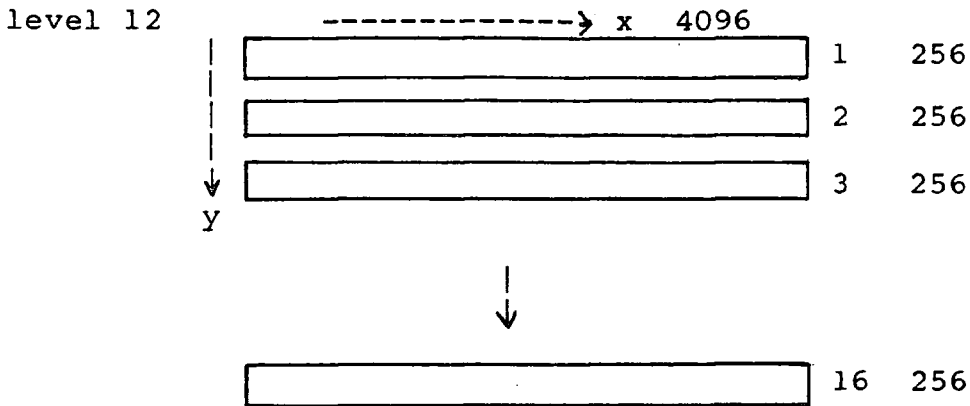
Mapping functions used in utility "QMCFORMAT"

See section 6.3.1 and annex 6a:

A description of the tape formats adopted for the transfer of OS raster data between Durham and Queen Mary College, London is given in section 6.3.1. A level 6 (64 x 64) index provides pointers to the associated 64 x 64 pixel DAP images of the level 12 (4096 x 4096) data. It has been necessary to develop mapping functions to provide these pointers between level 6 and level 12 raster representations. The level 6 data are stored in an array of 64 x 64 elements:



The level 12 data (4096 x 4096) are stored as sixteen partitions of 4096 x 256:



(a) Mapping function in x:

Each pixel address in the level 6 file must be mapped into a start and end address in the level 12 file. Each level 6 pixel represents a 64 x 64 cluster in the level 12 file.

let level 6 hit address of $x = p$
then level 12 start address in $x = [(p-1) * 64] + 1$
and level 12 end address in $x = 64 * p$

(b) Mapping function in y:

The theory for the mapping function in y is identical to that in x with some minor modifications. The y -addresses for

the level 12 file comprise sixteen partitions of 256 pixels (or 4 x 64 pixels) and hence it is necessary to compute:

1. which partition is desired and
2. the increment (1 - 4) of the collection of 64 scan-lines within the partition,

before the y mapping function can be determined.

```
let   level 6 hit address of y = q
      partition = Y = INT [(q - 1) / N] + 1
      partition increment = I = q - [(Y - 1) * N]
      where N = partition size in y/64
           = 4 in this case
```

```
then  level 12 start address in y = [(I - 1) * 64] + 1
and   level 12 end address in y = 64 * I
```

7 AN APPRAISAL OF RASTER-DRIVEN HARDWARE

7.1 INTRODUCTION

The digital manipulation of topographic data has taken on three distinct phases: capture (data input), manipulation/storage and presentation (data output). Investigations into the characteristics and efficient handling of the manipulation/storage stage are reported in chapters five and six. This chapter concentrates on techniques for appropriate mass data input and output by raster-driven hardware dedicated to this task. In particular, consideration is given to the quality and suitability of the data generated by commercially available raster scanning systems. These specialised devices are expected to have an increasing importance in automated map-production within the next decade. Although the cartographic market is somewhat limited, there are several commercial scanners available at varying costs and with a number of design strategies. Much literature has been produced describing each particular scanner (Harris et al 1981, Lloyd 1981, Pferd and Stocker 1981, Robertson 1980, Tafoya 1980, Rhind 1979), although little attempt has been made to provide any comparisons between each type of device.

Clearly, this is a deficiency; constructive appraisals of available hardware are crucial for effective system design and data base generation. The report by Davey et al (1977) is the only literature known to the author which has itemised and compared scanning systems; this document - commissioned by the Department of the Environment - gives a good summary of the state-of-the-art in the mid-seventies. Present indications suggest a rapid development in the field of raster data processing (Peuquet 1979a) and hence these documents are quickly outdated. A continual monitoring of the progress on a four - five year cycle is considered desirable. Further, until hitherto, no attempts have been made to encode digitally Ordnance Survey large scale data with scanning systems; provisional trials are crucial for a comprehensive assessment of the use of raster methods in handling topographic data in the UK.

A series of bench-marking trials have been carried-out with the kind cooperation of both manufacturers and users of raster-scanning equipment; these have enabled an investigation into the procedures and problems involved with this form for digital capture of OS large scales data. Table 7.1 itemises examples of scanning hardware known to the author and evaluations have been undertaken on those scanners asterisked - these are reported in section 7.3.

Table 7.1 Summary of ^{Some} existing raster-driven scanning systems

1. Bell Telephone Laboratories Inc.

Whippany, NJ, USA 07945

* EDAD - fibre-optic scanner - roller transport system

1016mm x unlimited length maximum document size
9.8 pixels/mm resolution
+ 0.01mm repeatability
+ 0.01mm accuracy
Grey-level detection
Raster-to-vector conversion software.

2. Broomall Industries Inc.

Broomall, Pa, USA 19008

LAS-200 - flatbed laser scanning system

1117 x 1625mm maximum document size
39.4 pixels/mm resolution
+ 0.025mm repeatability
0.05mm + 0.006mm accuracy
Grey-level detection
Raster-to-vector conversion software.

3. Kongsberg Ltd.

Kongsberg, Norway

* KartoScan scanner subsystem

600 x 1010 mm maximum document size
5-40 pixels/mm resolution
+ 0.01mm repeatability
+ 0.01mm accuracy
Grey-level detection
Raster-to-vector conversion software.

.....continued

4. Oxford University Nuclear Physics Laboratory

Oxford, UK

* PEPR - CRT film scanner

Documents reduced to 16mm or 35mm microfilm
1 in 10000 resolution
1 in 100000 accuracy
Grey-level detection
Raster-to-vector conversion software.

5. Scitex Corporation Ltd.

Herzlia B 46 103, Israel

* Response 250 - drum-scanner subsystem

914 x 914mm maximum document size
4-47 pixels/mm resolution
+ 0.01mm repeatability
+ 0.01mm accuracy
Twelve colour detection
Raster-to-vector conversion software.

7.2 SCANNING TESTS OF ORDNANCE SURVEY LARGE SCALE DOCUMENTS

The quality and formats of the source documents are expected to affect seriously any scanning processes; most scanners will accept both transparent and opaque media although image quality and contrast must be clear. There are, therefore, expected to be several problems associated with the accurate detection of some of the poorer-quality

master survey documents (MSD) currently in use at the Ordnance Survey, though similar difficulties will probably occur with the use of the automatic line-following equipment currently under-going tests in-house at Ordnance Survey (section 2.8). Unfortunately, it has been impossible to classify the extent of these problems without very substantial access to scanning equipment. In order to ensure the best data possible, at least for preliminary assessments, a specially prepared source document of a 1:1250 sheet (SO 9283SE) has been used in all the scanning trials. In the light of experience gained from this research, it is likely that most scanners will accept the higher-quality MSD's with little or no graphic enhancement. The extent of the modifications necessary to the poorer images is unknown at present; certainly some photographic enhancement will be necessary and possibly complete redrawing in (some) exceptional cases.

The raster information generated at the scanning stage will be a file of digital data representing the map image. Most suppliers of scanning hardware offer software options to transform this raster output into vector strings for further manipulation. General discussions with experts (Bell Laboratories, Broomall Industries, Kongsberg, Scitex, USGS pers comm 1980) prior to the scanning tests have suggested that the performance of this conversion process will be



Figure 7.2: Extract of special OS
source document used for scanning
trials - text and stipple removed.
Crown Copyright Reserved.

seriously degraded by the textual information and the building infill (stipple) included in the OS large scale maps specification. Text and feature infill can be stored successfully in the raster-files but cannot be vectorised and described succinctly as vector strings by existing software. Thus, further development in the raster-to-vector conversion methods is necessary before an all raster-driven OS map production system is possible. For these reasons, the specially prepared source document in the scanning trials had both text and stipple information removed. An extract of this source document is shown in figure 7.2.

High-quality copies of this graphic on stable base photographic film were sent to the following raster-sites with a request to scan the image at a resolution considered most appropriate for accurate digital representation.

1. Bell Telephone Laboratories Inc.
2. Broomall Industries Inc.
3. Kongsberg Ltd.
4. Oxford Image Analysis Group.
5. United States Geological Survey (USGS) - operating a

Scitex Response 250 system.

All systems operational at these sites perform an immediate conversion to vector format for most mapping although requests were made for both the raster data and the vector data to enable comprehensive assessments of the degrees of success from both the systems' hardware and software.

7.2.1 The role of a raster-driven data entry system

Scanning devices provide an interface between each source graphic and the computer system. The role of this specialised hardware is to accept the existing plans (or some photographically reduced equivalent) and resolve the line-work into its constituent pixel data. There is a limited number of design strategies for this task - they involve either attaching map sheets to moving or fixed flatbed frames, to drums that rotate at a controlled speed or use friction rollers to transport the document through a fixed scanning system. Regardless of the mechanics of the device, however, the role of each device is effectively the same. An increment in the y-direction creates a new scan-line and a series of contiguous x-shifts establishes the

pixel information along each scan-line. With the exception of the Scitex Response 250 Super-Scanner which can distinguish up to twelve colours, most systems can classify only monochrome information. Each pixel comprises a measure of the greyness of a very small element of the map image. For example, assuming eight-bits are allocated to each raster pixel, then the grey-levels are held in the range 0 to 255. Areas of no data (pure white) are represented by a zero and black areas in the centre of map line-work are coded as 255 or values near to it. Pixels which fall on the edge of lines will comprise values somewhere within this range; 'noise' arising from such diverse factors as dirty source graphics or poor signal-to-noise ratios of the scanning equipment will also have intermediate values.

The resulting digital file is a large array of pixel data representing the original map. Those pixels with high grey-levels (darker regions) are associated with the map line-work and pixels with lower levels of greyness represent the lighter areas which appear as empty spaces on the map sheets.

In line mapping (as distinct from encoding continuous tone images) it is necessary to assign a threshold grey-level to demarcate the boundary between black and white pixels. This will be assigned empirically by a trained operator and can

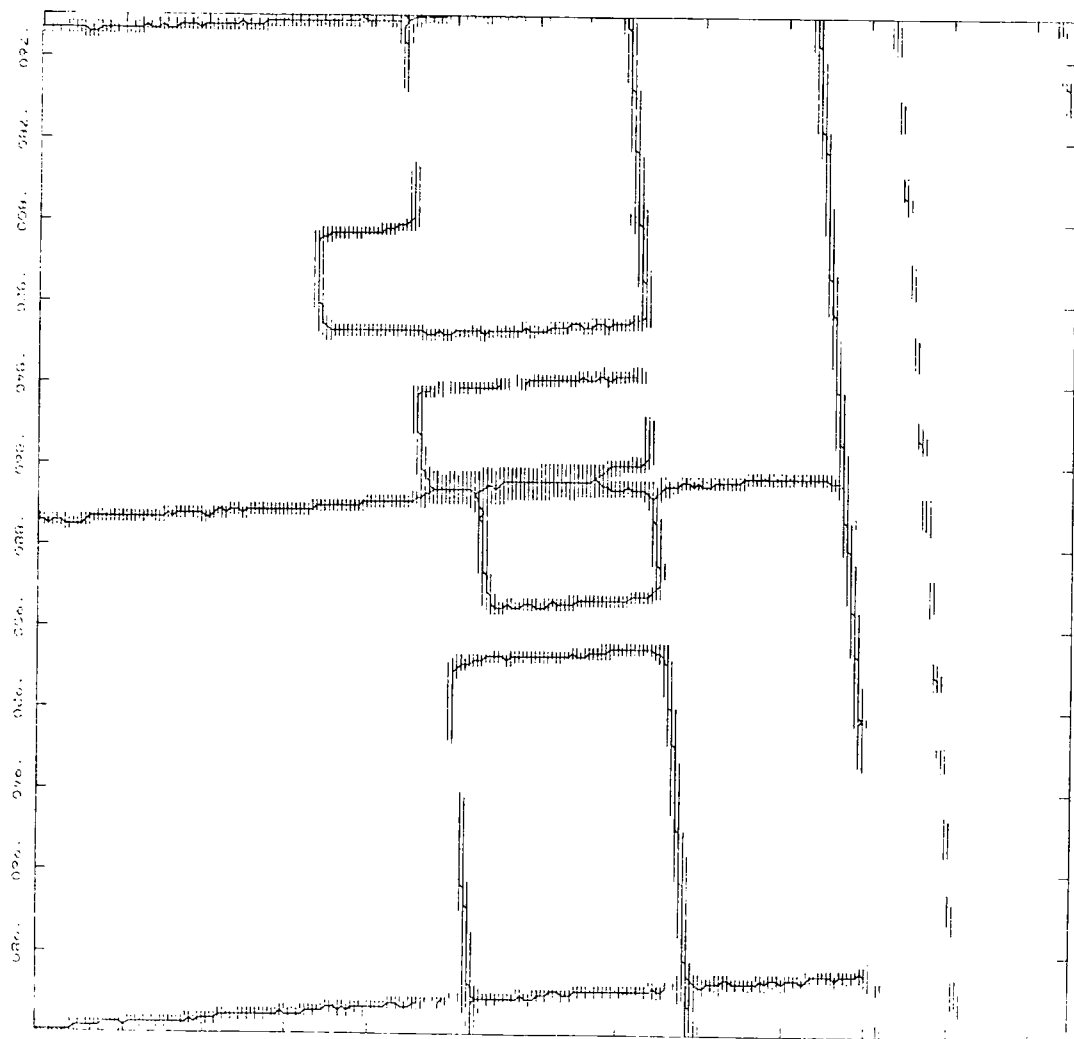


Figure 7.3: The effects of incorrect grey-level thresholding with OS map data in the Oxford Univ. PEPR system.
Crown Copyright Reserved.

vary from document to document depending on image quality and data characteristics. The tests carried-out in this research have, by necessity, been constrained by the thresholds chosen at the scanning sites. It has had to be assumed that these choices are adequate for suitable digital representation of the OS data. Clearly, the choice of grey-level threshold can be more accurately defined by a trained operator as a result of experience and experimentation than by the researcher unfamiliar with the equipment. The effect of the threshold level is to tune the scanner to detect lines of a desired line-weight. Varying the threshold will affect the pixel-width of the vectors in the raster image: a low threshold will provide line-widths comprising many pixels since more low-order grey-levels are accepted. Figure 7.3 shows the effect of an inappropriate threshold setting at the scanning stage. The close proximity of parallel lines are seen to merge together, producing errors in the raster file and in any subsequent vectors.

This has summarised the role of any scanning subsystem in the total raster process; further data manipulation is not dependent on the scanner and is performed post hoc by computer software. Detailed accounts of the nature of this software is given in chapters four, five and six.

7.3 DESCRIPTIONS OF COMMERCIAL RASTER-SCANNING SYSTEMS

There now follows a brief overview of raster scanners known to the author which are considered appropriate for handling cartographic data.

7.3.1 Broomall Industries Inc.

Broomall Industries Inc. (BII) offer a complete raster system - 'SCAN-GRAPHICS'. This includes scanning hardware, controlling PDP 11/70 mini-computer, editing facilities and all associated software. Early scanning subsystems used white-light sources to detect the raster pixels: this was found to be error-prone and difficult to implement (Broomall pers comm 1980). More advanced hardware using laser-optics has proved to be more accurate and reliable.

There are currently two scanning systems commercially available from BII (Broomall Industries Inc. 1980): the LAS-200 is a high resolution flatbed laser scanner and the D-400 is a more mediocre, lower resolution drum-type scanner. The latter is intended for small-format graphics - engineering drawings, artistic line-work and so on. It does not have potential for Ordnance Survey production methods without major modifications to the scanning source documents.

The maximum size acceptable is 12 x 18 inches which is too small for any serious cartographic work. Nevertheless, the D-400 is a compact (38 x 25 x 11.5 inches), versatile scanner offering a scanning resolution of ten pixels per millimetre and a minimum line-weight detection of 0.1mm.

The Broomall LAS-200 flatbed scanner fits more neatly into a cartographic production environment; it has a technical specification (Table 7.1) quite satisfactory for Ordnance Survey mapping. The large-format scanning area will enable up to four large scale plans to be scanned during each machine set-up. The laser scanning head has the ability to speed-up and slow-down as data complexity varies and hence will run proportionally quicker for the sparsely covered 1:2500 rural data (Adams and Rhind 1981) than will be the case for the more detailed 1:1250 plans. A disadvantage for production use is the large dimensions necessary for the flatbed scanning easel, producing a device which requires much space. Each unit measures 115 x 86 x 32 inches making it comparable with the large Ferranti Master plotters presently in use at the OS headquarters.

Early systems emanating from BII - where the raster data are stored on disk temporarily for post hoc vectorising - have been superseded by a more impressive real-time vectorising strategy. BII claim (Broomall pers comm 1980)

that data in a raster form are difficult to store, manipulate and edit (cf. alternative to Scitex ideas - 7.3.5), and that the elimination of any raster storage provides for a more efficient system. The detailed workings of the algorithm are a commercial secret although some general features can be reviewed. The LAS-200 device scans the graphic document in two-inch swathes, generating raster data for each swathe. A microprocessor buffers the data and transfers them to the controlling mini-computer. Once the first swathe has been completed, the vectorising software begins the raster-to-vector conversion whilst the second swathe is being scanned. Vectorising speeds in the range 5000 to 10,000 lineal inches of arc per hour are claimed (Broomall Industries Inc. 1980) although intuitively it is expected that Ordnance Survey data will be far slower than this. Although no feature-tagging is possible during the scanning or vectorising stages, the software has the ability to detect closed polygon data and flag each feature by line-weight. Approximate costs for a Broomall system (LAS-200, mini-computer, software) are \$300,000. A vector-driven interactive editing station (with software) for post hoc data validation, feature-tagging, etc. costs \$80,000 (Broomall pers comm 1980).

7.3.2 Bell Telephone Laboratories Inc.

The Whippany NJ base is responsible for providing and maintaining all utilities connected to Bell Laboratories' plants. In this task, it is predominantly concerned with technical and engineering drawings. In 1975 a study was initiated to investigate the cost-effectiveness of using digital techniques for the storage and manipulation of these drawings; this has led to the initiation of a development project for a scanner to perform the graphic-to-digital conversion process (Pferd and Ramachandran 1978). Effective scanner design was constrained by the following factors:

1. it had to be inexpensive;
2. it must be able to handle large graphics - typically E-size (30" x 40") and K-size (36" x 60").

For these reasons, any form of lasers, flatbed scanners or drums were rejected. Further, the nature of the engineering drawing data has dictated an immediate conversion to a vector form to reduce data storage overheads and increase processing rates. After some four years of research, a working system has been developed which meets their scanning requirements and can distinguish between different line-weights - an

important necessity for engineering drawings (Pferd and Stocker 1980). The device operates with a linear-grid of optical fibres which transfer a reflected illuminating light source from the graphic to a video camera and on to a video-to-digital converter. A pair of friction rollers, controlled by a servo-motor, provides the transport system for the graphic whilst scanning is taking place. At best, the system can provide a scanning resolution of 250 pixels per inch (9.8 pixels per millimetre) - this is a function of the diameter of each fibre-optic and is an acceptable resolution for engineering drawings.

The system operating at Bell Laboratories is coupled to a time-sharing PDP 11/70 processor with 256k bytes of memory, supporting the UNIX operating system with up to 100 simultaneous users - most other commercial scanners require a dedicated mini-computer system to operate efficiently. It is the aim of the Bell Laboratories to produce a computer peripheral for about \$25,000 which eventually will be as commonplace as a Tektronix graphics terminal. This will enable fast local digitising for remote copy, teleconferencing and editing (Pferd et al 1979). One scanner for every Tektronix 4014 is foreseen in the future.

In order to reduce data volumes, thereby increasing processing and data broadcasting rates, the real-time

vectorising software has been developed to run simultaneously with the scanner. Pferd and White (1981) describe the nature of this software and the coding strategies adopted for the Bell Laboratories' requirements. Generally, the system is not intended as a cartographic tool; resultant vector output is optimised for a teleconferencing system (i.e. rapid conversion to vectors to give an approximation of the image being scanned). Nevertheless, the system provides a clear indication of the nature and potential of inexpensive scanners for the medium-term future. Although the vectorising software is inappropriate for most cartographic work, the scanner can provide an operational data entry system for more appropriate post hoc software - for example, the ideas using parallel-computers discussed in chapter six.

7.3.3 Kongsberg Ltd.

Kongsberg Ltd. of Norway have traditionally been much respected for their high-quality vector plotting devices. More recently, in cooperation with Messerschmitt-Bolkow-Blohm (MBB) of Munich, they have developed the SysScan automated mapping system which is aimed towards the large-scale utility, topographic and hydrographic market (Lloyd 1981). In essence, the SysScan system comprises three components: raster scanning data entry, a processing/interactive editing

facility and data presentation.

The KartoScan scanning subsystem provides the means of automatic data entry although additionally the system can accept data from other sources: Total-Station surveying equipment, photogrammetric devices, manual digitising and so on (Kongsberg Ltd. 1980). The scanner operates off-line and is connected to a magnetic tape drive to provide an interface with the SysScan controlling mini-computer - a PDP VAX 11/780 supporting the DEC VMS operating system. The source documents (which can be transparent or opaque) are held to a flatbed easel by glass pressure and are scanned in a series of swathes (strip by strip) by the optical scanning head. There are 507 pixels per swathe; each swathe is aligned to its neighbours without leaving any gaps or overlapping. The width of these swathes and the time taken for a complete traverse of the source graphic is dependent on the selected scanning resolution. Table 7.4 summarises the characteristics of each swathe for the four possible scanning resolution options.

Table 7.4: The effect of scanning resolution on pixel swathes in the Kongsberg KartoScan

Source: Kongsberg Ltd. (1980)

<u>Pixel resolution (mm)</u>	<u>Width of swathe (mm)</u>	<u>Traverse rate in mm/sec (1600 bpi tape)</u>	<u>Theoretical max scan times for 1:1250 map (mins)</u>
0.2	100	32	1
0.1	50	16	4
0.05	25	8	14
0.025	12.5	4	54

The scanner provides a file of grey-level pixels in the range 0 to 127 which are transformed into an equivalent bit-map representation (black/white) with an appropriate threshold boundary (section 7.2.1). Once in a suitable form, the data enter the Kongsberg data base GINIS (Graphical Interactive Information System). This data structure is hierarchical in concept and takes on a tree-like representation (Lloyd 1981, Martin 1976). Each branch in the tree comprises references to a specific collection of like-attributes in the map image. The parent to each branch provides a summary of its children. For example, a branch comprising references to all buildings in a map image is the parent of sub-branches which comprise pointers to private buildings, public buildings and so on. GINIS is a

vector-driven system providing a means of editing and manipulating topographic data easily. Clearly, an intermediate processing stage is necessary to reorganise the binary scanned data into specific entities; buildings, for example, must be constructed from specific lines and through recourse to adjacent text information. The Kongsberg system provides two mechanisms for this operation: one operates automatically and the other involves large amounts of human interaction. During the conversion stage between raster and vector, each resulting line is 'tagged' with a description of its line-width on the original document. In this way, specific lines can be extracted from the bulk of the data automatically and held at some level in the data base hierarchy. Since line-widths vary only slightly in OS large scales mapping, then this facility is expected to offer little assistance for entity re-building in production. It is the second mechanism available which is expected to offer most potential for production at present, although such an approach may prove too costly to be cost-effective. This second option involves the use of the Kongsberg Graphics-7 (Kongsberg Ltd. 1980) interactive graphics screen with associated light-pen cursor. With such hardware and the Kongsberg controlling software, it is trivial to point to features on the screen and classify them quickly into particular categories to take advantage of the flexible storage and retrieval mechanisms offered in the GINIS data

base. Lloyd (1981) summarised the facilities offered by the complete Kongsberg system and provided an indication of the design of the data structure used in GINIS.

The presentation stage of the SysScan system incorporates small electrostatic raster plotters (see section 7.5.2) for quick display or high-quality traditional Kongsberg vector-plotters for accurate work.

Each component in the SysScan system is sold separately to enable customised installations to be purchased. In mid-1981, Kongsberg Ltd. quoted the following prices for their SysScan units. Data entry subsystem (scanner, magnetic tape drive, vectorising software) - £220,000. Data editing/archive subsystem (VAX 11/780 computer, terminal, 67M byte disk drive, magnetic tape drive, GINIS software) - £ 200,000. Data presentation subsystem (high-quality flatbed plotter) - £ 80-90,000 (Kongsberg Ltd. pers comm 1981).

7.3.4 Oxford University Image Analysis Group

Since the early nineteen seventies, two precision CRT film scanners have been in use at Oxford University for several image processing applications; both were originally developed for bubble chamber film processing tasks (Davey et al 1979).

The work of the Image Analysis Group has been reported widely (Harris et al 1981, Black et al 1979, Davey et al 1979, Davey et al 1976) and has involved much progress in the automatic interpretation of line-structured data.

Source documents to be scanned are first photo-reduced to 16mm or 35mm film for entry to the PEPR scanner. The hardware will execute autonomously a single sweep of the cathode-ray through a distance of 500 microns with a parameterised threshold from a software-controlled origin. The signal processing returns pointers to positions where the detected threshold on the scanned image is higher than the given sweep threshold. Unlike the majority of scanners, scanning on the PEPR as a series of successive parallel sweeps is generated under software control by repeatedly defining the sweep origin.

The PEPR scanner is not completely suitable for cartographic work; there are two major defects in its specifications. One arises from the sensitivity of threshold variations in the photographic image causing gaps in the resultant pixel data - such an occurrence is commonplace with scanning devices operating to micron standards. The second is the large amount of noise introduced into the pixel data largely a result of the photographic preparation stage and the signal-to-noise limitations in the PEPR (Preston pers

comm 1981). Similar effects would be evident, for example, from a poor-quality source document being scanned on a high-quality scanner. Current research at Oxford is attempting to develop pixel-preprocessing software to 'clean-up' poor quality data prior to the post hoc vectorising stage (Preston pers comm 1981).

7.3.5 Scitex Corporation

The functional role of most of the raster scanning systems examined thus far has been for rapid data entry (in pixel form) with immediate conversion to a vector notation for storage and other processing. The Scitex Corporation of Israel have adopted alternative strategies; they consider raster data capture and subsequent raster storage and manipulation to be the most flexible approach for handling cartographic data. The Response 250 system was released in the late-seventies primarily for the map-production industry; it is an extension of the Response 200 system introduced in 1975 for textile and decorative printing applications (Gronemann and Bar-Lev 1979). The Response 250 comprises three components (cf. Kongsberg Ltd.): scanner, colour editing station and laser plotter (Scitex Corp. 1981). Each subsystem requires a central-processor to run independently; the controlling computers are Hewlett Packard - the central

processor is a HP 24MXE with 32k words of 16 bit memory with an attached tape drive and a 120 megabyte disk drive.

The basis of data capture is a large format (90 x 90 cm) drum scanner on to which is placed full-size opaque or transparent originals held by vacuum and pin-registration marks. Single-pass scanning is carried-out in monochrome or in up to twelve colours or hues. Pixel resolution can range between 4 and 47 pixels per millimetre and is set by the operator; the data are run-length-encoded by the scanning hardware and stored directly on magnetic disk.

The interactive edit station comprises a nineteen-inch colour display with refresh memory of 320 x 256 pixels of four bits, a digitising tablet and cursor and a series of function buttons for the most common manipulation tasks. Functions include: line/curve drawing, area infill, zooming, feature movement, orientation changes, colour or feature recoding, feature deletion and so on. The interactive capabilities are impressive and fast, largely a result of the efficient use of data compaction methods and the dedicated computing facilities. Text and name placement is treated in exactly the same manner as point symbols. Text characters and user-defined point symbols are held on disk in a pattern library for easy recall and interactive placement in the digital image. The magnetic tape unit provides the means of

transferring data between the components of the Response 250. It is also possible to input satellite imagery data into the system and perform interactive editing and image enhancement and display as though they had originated from the scanner.

Hardcopy data presentation and display is possible by a number of means. The laser colour plotting subsystem provides a high-quality raster display for large format graphics (up to 185 x 100 cm). Plotting resolution is variable up to 72 points per millimetre; plotting speeds are dependent on this and on the characteristics and density of the data and the plot dimensions. Vector plotters can display linear features which have been converted from raster form by software.

Costs of the Scitex hardware in mid-1981 were c. \$250,000 for each component including the controlling computers. In essence, a full system can be purchased for approximately \$750,000.

7.3.6 Other scanning devices

Sections 7.3.1 to 7.3.5 have summarised the commercially available raster hardware known to the author. A number of other devices have been examined which do not currently match

cartographic requirements although close monitoring of their progress is advised since common problems and solutions are bound to develop. Some of these systems are outlined below.

GEC-Marconi Ltd. of Chelmsford are using raster-driven techniques in a project to study the problems and characteristics of handling information originating from a synthetic aperture radar (SAR) device. A 'one-off', specially designed scanner/camera has been developed to effectively represent the SAR in laboratory conditions. The camera is capable of resolutions of 13 microns and generates 1024 x 1024 pixels per image; each pixel is not square - measuring 13 x 12.5 microns in dimension. The graphic to be scanned must be no larger than A4 format and is secured to a flatbed easel above which is placed the camera which scans in the x-direction under servo-motor control. A quartz-halogen light source is reflected off the graphic and detected by the camera which transmits its analogue signal through an analogue-to-digital converter before being stored on floppy disks. The source graphics must be of high-quality; provisional trials have included TV test-card data and aerial photographs (Blythe pers comm 1981).

Levels of noise are intolerably high for most cartographic work; this is largely due to such fine scanning resolutions. The system incorporates up to 1000 grey-levels and is

inefficient in storage - 24 bits are allocated to each pixel. In essence, a 1024 x 1024 image requires in excess of three megabytes (8 bits per byte). Primarily for reasons of speed, there are no facilities to filter or modify the incoming data as scanning is taking place; each digital image generated exceeds the capacity of the floppy disks. The dual-density volumes in use (c. 0.5 megabytes capacity) will each store only a one-centimetre partition of the original image. Visual inspection is possible as scanning is in progress by sampling the four most significant bits of each incoming pixel and displaying them on a TV monitor. The research project is technologically intensive and well in advance (they claim) of similar work in the world although, generally, the nature of developments are not considered suitable for any Ordnance Survey applications without much modification and further research.

Raster-driven hardware, generally, has become increasingly popular in a number of industries - some are partially related to map production. In particular, interactive plate-making in the colour-printing industry has gained importance as appropriate systems have been developed. A typical device is the Magnascan 550 (Crosfield Electronics 1981) which has an architecture similar to the scanning hardware marketed by the Scitex Corporation. Though discussed by Crosfield and the ECU in circa. 1970 (Rhind pers

comm 1982), there are no printing systems of this nature known to the author which offer facilities to extract the digital information for other types of manipulation. Their applications are restricted to graphics editing to produce screening, continuous tone, positive or negative separations and so on; any intermediary digital processing is concealed completely from the operator who has no control on the system software. There is much to be said for developing suitable interfaces to these devices for extended facilities. Most large map-producing companies are already considering their use in reprographic operations (Fortesque pers comm 1981) and hence they can provide a means of mass data entry into an interrelated computer data base.

The list of raster-dedicated systems described in the previous sections is not exhaustive although these are the major examples studied by the author. Current indications have suggested that the next five-to-ten years will see the introduction of others (probably based on similar strategies) which are cheaper and more versatile.

7.4 AN APPRAISAL OF EXISTING RASTER SYSTEMS

Section 7.2 itemises the suppliers of scanning systems who were invited to participate in the bench-mark trials to scan

a specially prepared high-quality OS 1:1250 map sheet (figure 7.2). The cooperation from these suppliers was excellent; with the exception of Broomall Industries, all agreed to provide samples of data generated by their systems. Section 7.4.1 describes the results obtained from the scanning hardware and section 7.4.2 considers the nature and quality of the resultant vector data generated by software. Bench-marking tests on a Scitex Response 250 system were undertaken with the kind cooperation of the USGS who operate a system in their map-production flowline. In mid-1980, when the project was initiated, Scitex Corp. had no bench-marking facilities.

The results:

The output tapes from each scanning system differ markedly, reflecting the wide diversity of data formats in common usage with systems effectively performing the same task. Much work was necessary to decipher the incoming tapes; this ranged from low-level assembler programming to unpack bit-compacted pixels from the USGS to translating German language format specifications supplied by Kongsberg (an indication of their close association with MBB in Munich - see section 7.3.3).

Table 7.5 Summary of the characteristics of the raster-files
generated by four tested scanning systems

	Bell Labs EDAD	Kongsberg KartoScan	Oxford U PEPR	USGS Scitex Response
Scanning resolution (mm)	0.05	0.10	0.06	0.05
Raster dimensions	8000 x 2000	4000 x 4000	6441 x 6385	8000 x 7979
Bits per pixel	see text	8	8	1
Sparseness of file (%)	94.66	94.12	-	94.45
Total tape length (feet)	99	1219	-	848
Recording density (bpi)	800	1600	-	1600
Machine set-up time (mins)	c.15	10	-	15
Scanning times (mins)	unknown	15	unknown	72

No detailed information can be given on the characteristics of the raster file generated by the Oxford PEPR system as only the vector data were supplied.

7.4.1 The characteristics of the raster files

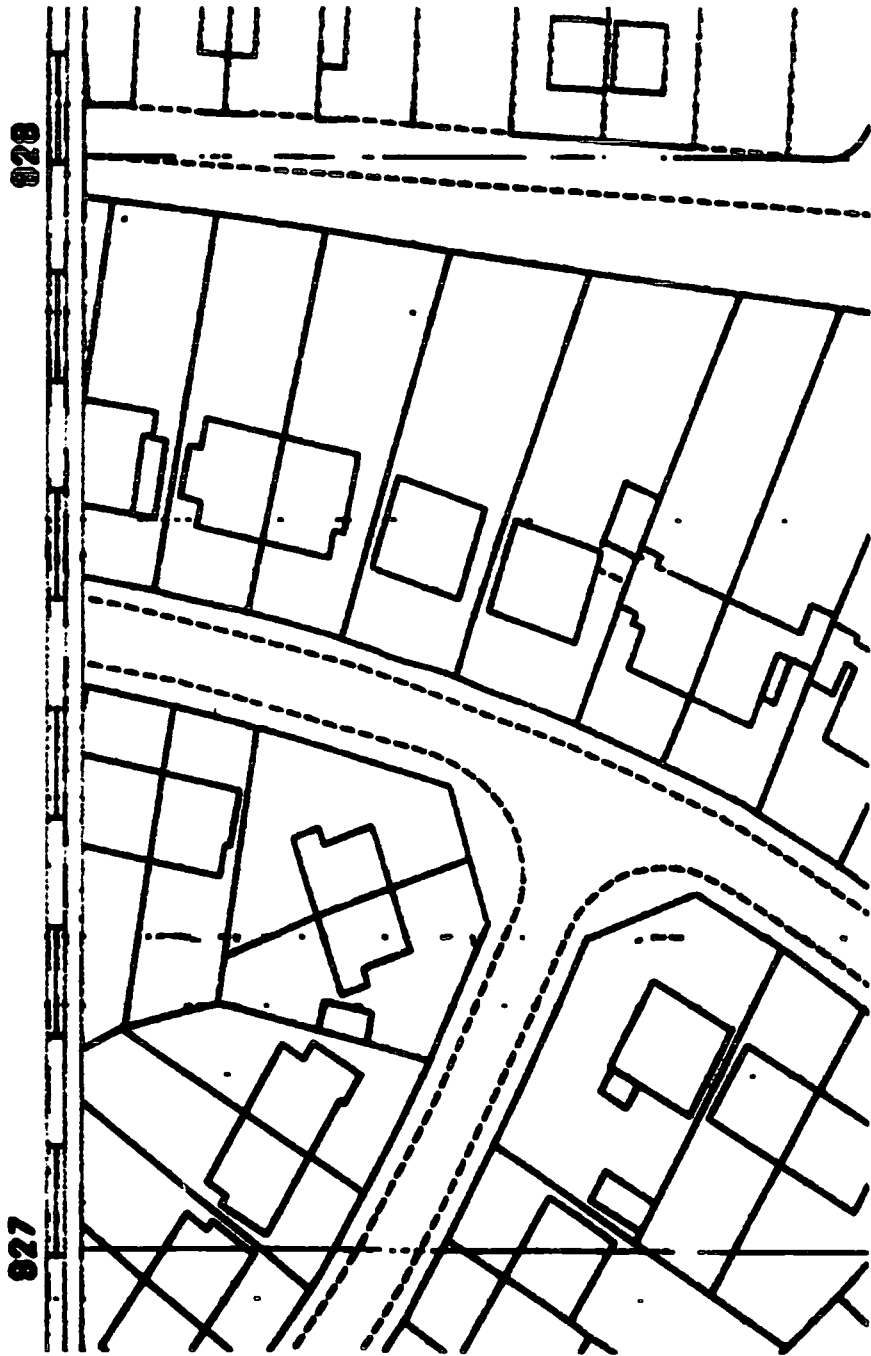
Comparisons between the digital raster files generated by each scanning system is difficult because pixel resolutions, storage notations and formats differ markedly. In future trials of scanning systems by the OS, it would be better to request a specific single pixel resolution - 0.05mm is desirable for appropriate graphics and other manipulation. Nonetheless, on the basis of the work carried-out in this research, it has been possible to compile table 7.5 which itemises the important characteristics of each raster file produced.

The raster-file from Bell Laboratories is not a complete representation of the whole map sheet. The fixed scanning resolution of 250 pixels per inch of the EDAD scanner was considered too coarse for accurate representation of the map image. A central strip of the document was photographically enlarged by a factor of two prior to scanning in order to give an increased resolution of 500 pixels per inch (0.05mm). The raster file - comprising 8000 x 2000 pixels - represents one-quarter of the total map-sheet. Each scan-line in the file is run-length-encoded (see section 5.4); only the start and end addresses of the set-pixels are stored for each scan-line. A resume of the Bell Laboratories tape formats is given in appendix 7.1 to provide an example of a commercial

Figure 7.6a: Extract of raster image
generated by Bell Laboratories Inc.
EDAD scanner. Crown Copyright Reserved

ORDNANCE SURVEY

80 9283 NE



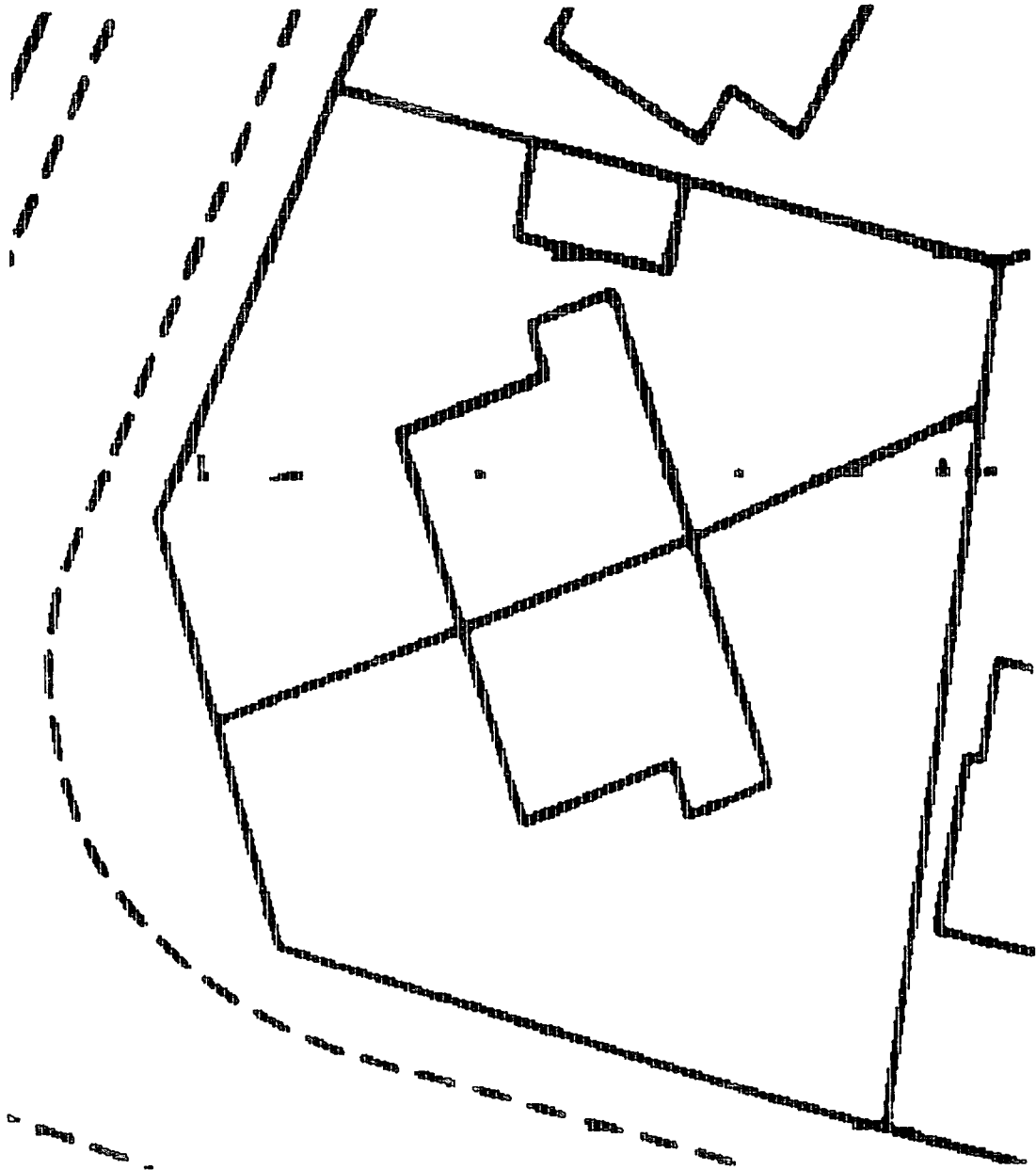


Figure 7.6b: Enlarged section of
raster image generated by Bell Labs.

EDAD scanner.

Crown Copyright Reserved.

run-length-encoding scheme. Figure 7.6a,b shows a graphical representation of the raster file prior to the real-time vectorising stage in the Bell system. There is an inherent scale distortion in the images due to each pixel being rectangular; the vertical step-size is approximately 1.075 times the horizontal step-size. Most non-graphic processing can compensate for this deficiency although the modifications of the data for graphics is more difficult.

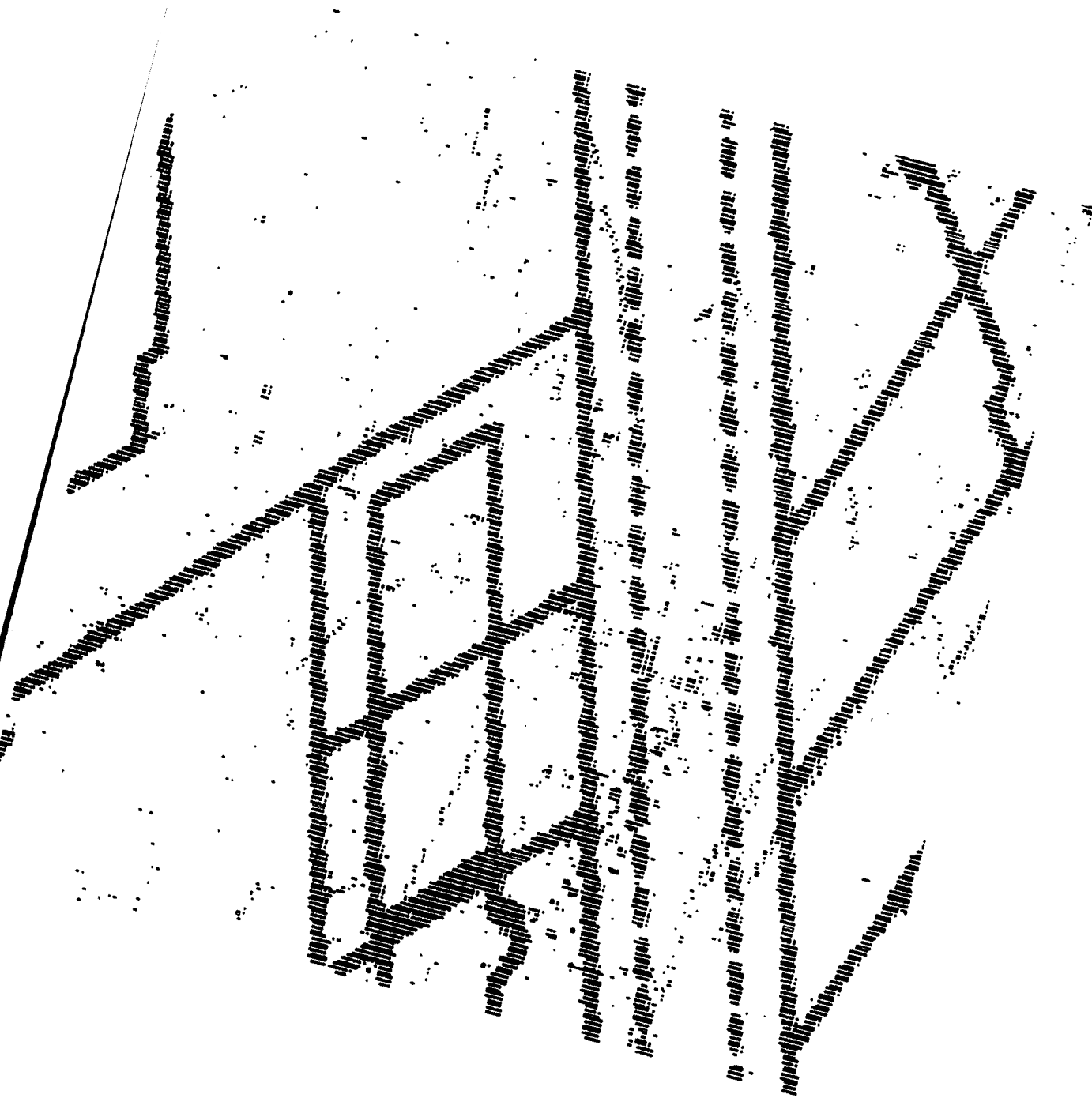
The evidence of missing line-work (especially on the narrower-gauge grid-lines) is a function of the choice of a detection threshold described in 7.2.1. To lower this threshold would introduce the missing line-work but also generate more extensive levels of 'noise', affecting image quality, and post hoc vectorising. The extent of the noise in the raster file is clear in figure 7.6; in particular, two 'noisy streaks' (lines) are apparent at approximate grid locations 927.3 and 927.7. It was stated by Bell Labs. that these were caused by broken fibres in the scanner. There are other randomly dispersed areas of noise in the image, which appear as clusters of set-pixels - usually of the order of 8 x 8 pixels wide. This may be due to dust particles/scratches on the original, or poor signal-to-noise levels in the scanner. In general, the levels of noise are not excessive given the low-cost of the device and the consequent relatively poor signal-to-noise ratio. Most software can

eliminate the effects of noise easily, although areas where it is attached to valid line data may be problematic in further data processing. This noise appears as bulges in the weight of lines in the the map data (see figure 7.6c).

No scanning times have been given for these trials and hence it is difficult to comment (with certainty) on the efficiency of the Bell system from a production standpoint. Nonetheless, it was pointed-out by Pferd (Pferd pers comm 1980) that the scanner performs quickly and is not considered a serious threat to the demise of a production system.

The format of the raster-file generated by the Kongsberg KartoScan system reflects the scanning strategy performed by the hardware (section 7.3.3). The raster data are held as a series of grey-levels (in uncompactd form) sequentially - swathe by swathe (table 7.4). Each logical record comprises 512 bytes; there are five bytes of header information for each record, followed by a single scan-line of 507 pixels (with one grey-level per byte). Figure 7.7 shows the Kongsberg raster-file graphically. The apparent high level of noise is a function of the detection threshold described in section 7.2.1. It has been necessary to allocate a threshold value to distinguish the boundary separating black from white in the grey-level data. As outlined in section 7.2.1, there are no formal rules dictating a threshold level

Figure 7.7: Extract of raster image
(grey-levels) generated by Kongsberg
KartoScan scanner.
Crown Copyright Reserved.



- this is derived empirically and varies from graphic to graphic.

The scanning resolution adopted for the tests by the Kongsberg operator was 0.1 millimetres although the hardware is capable of scanning to finer resolutions if desired (typically 0.05 and 0.025 millimetres). No preparation stages were necessary for the source graphic; the KartoScan can handle the full-sized original drawing. Kongsberg supplied time estimates associated with these tests: the total time was quoted to be twenty-five minutes. Initialising the scanner (magnetic tape loading, source graphic loading, machine calibration) took ten minutes and actual scanning took a further fifteen minutes. It is clear that this scanning time quoted does not agree with the theoretical figures obtained from the KartoScan technical documents and shown in table 7.4. Further communication with Kongsberg Ltd. (Lloyd pers comm 1982) suggested that an additional time period must be included for such factors as movement of the scan-head to the start of each swathe; further, it was claimed (by Lloyd) that additional times are always quoted since operational scanning rarely meets the theoretical maximum speeds suggested by the technical specifications. It is clear that one should be wary of commercial propaganda in technical specifications and any further scanning trials must set-out clear objectives of

required quality and processing times to be met.

The quality of the results attained from the Kongsberg KartoScan scanner were difficult to assess without adequate access to an appropriate raster plotter. To hold the data as a series of grey-levels requires repeated plotting with varying thresholds to determine the value most appropriate to distinguish black and white. Nonetheless, on the basis of a random selection of areas in the raster file being displayed with varying thresholds on a Tektronix cathode-ray terminal, it is evident that the KartoScan produces a reasonable digital representation of an OS large scale source document. There is also evidence of substantial amounts of 'noise' in the data which must be cleaned post hoc by software. Careful threshold calibration at the setting-up stage is essential for accurate depiction of the data in digital form; apart from the presence of noise, there is also evidence of omissions of thin lines (e.g. grid-lines) or data blocking of dense line work and regions with very close parallel lines. It is clear that the post-scanning software being developed by Kongsberg is tuned precisely to the output of the KartoScan; great difficulties were encountered in processing the Kongsberg raster data with home-written software by the author.

The raster-file generated by the Scitex Response 250 at the USGS comprises a bit-serial representation of the scanner's output and hence there have been no problems with the selection of an appropriate threshold for display purposes. The scanning resolution is twenty points per millimetre, generating (theoretically) an array of 8000 x 8000 pixels for the complete 1:1250 map. In practice the scanner was halted prematurely in the x-direction producing a file of 8000 x 7979 pixels. Such a bit-serial pixel representation provides an efficient method of transferring the Ordnance Survey monochrome raster images between computer installations. Each logical record of 1000 bytes (8000 bits) comprises a single raster scan-line. An IBM assembler routine has been written to unpack the bit-serial data into one-byte units for easier manipulation with FORTRAN.

The quality of the results attained from the USGS Scitex scanner are encouraging for Ordnance Survey applications. Examples of these data are shown in figure 7.8a,b,c. A clear, error-free image has been generated with no visual evidence of noise or feature omissions. Since no skeletonisation had been carried out at this stage, the width of each line (number of pixels) is a function of feature orientation although in general they are approximately six pixels wide.

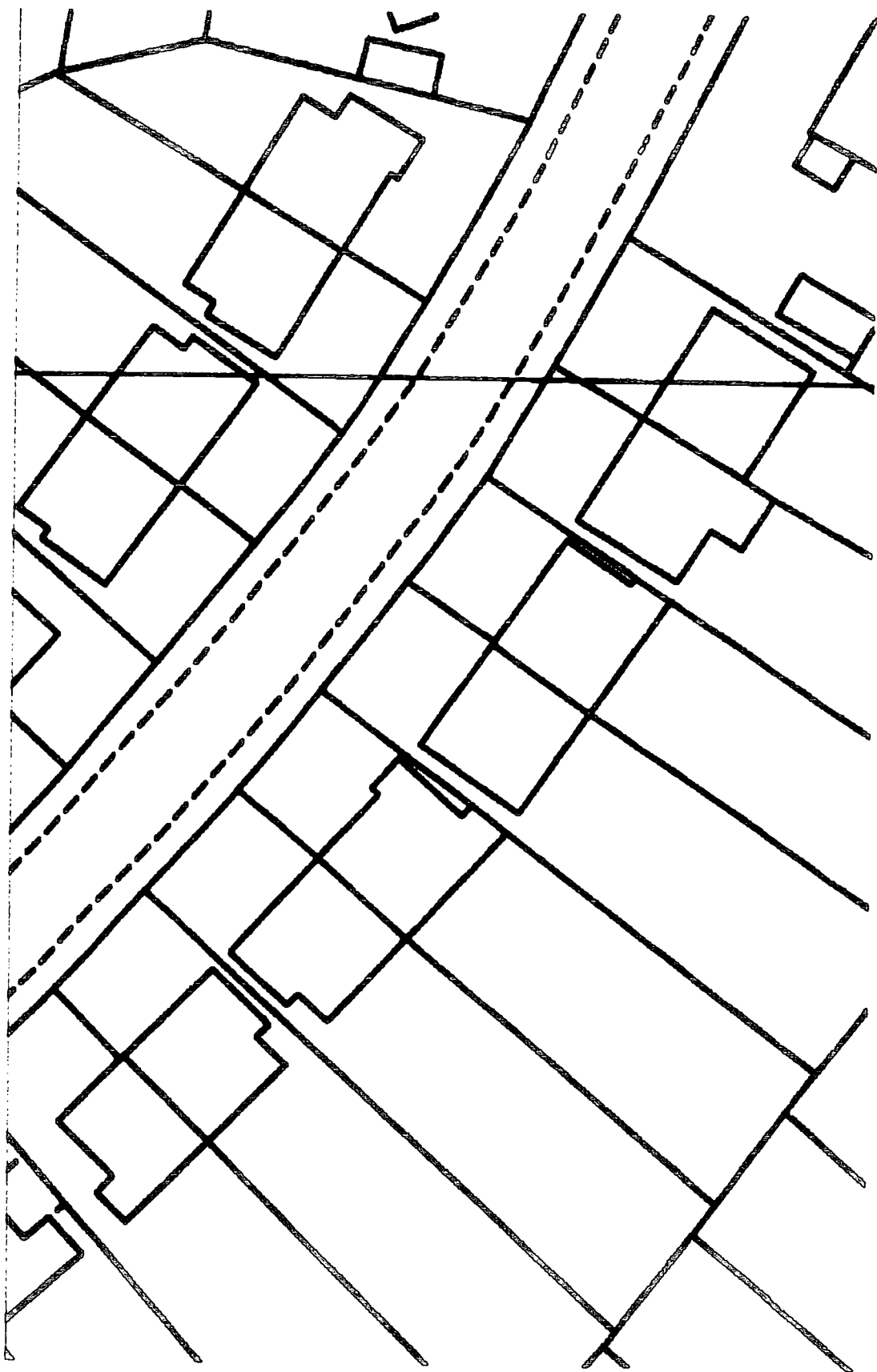


Figure 7.8a: Extract of raster image
generated by Scitex Corp. scanner.
Crown Copyright Reserved.

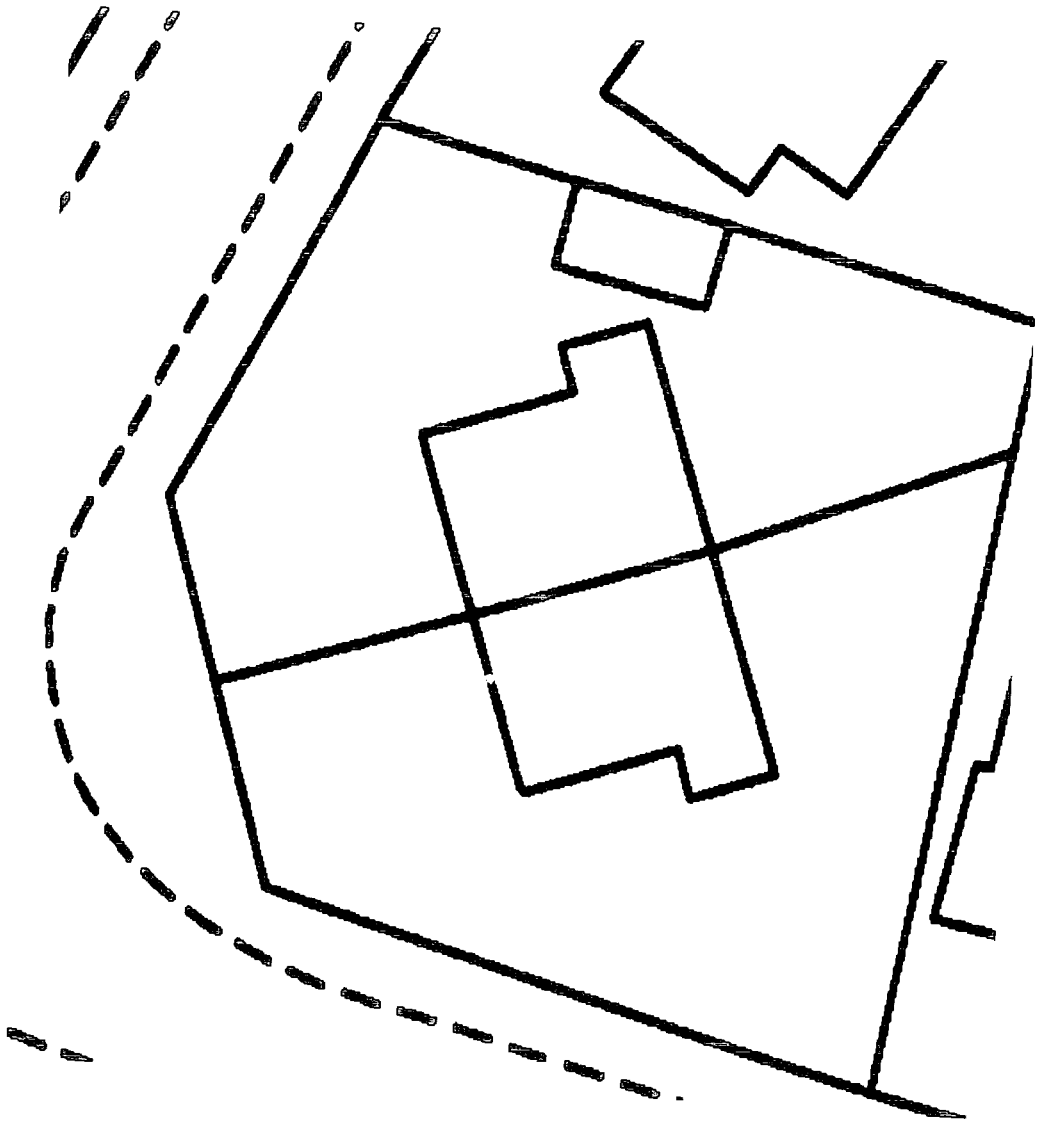


Figure 7.8b: Enlarged section of raster image generated by Scitex Corp scanner.

Crown Copyright Reserved.

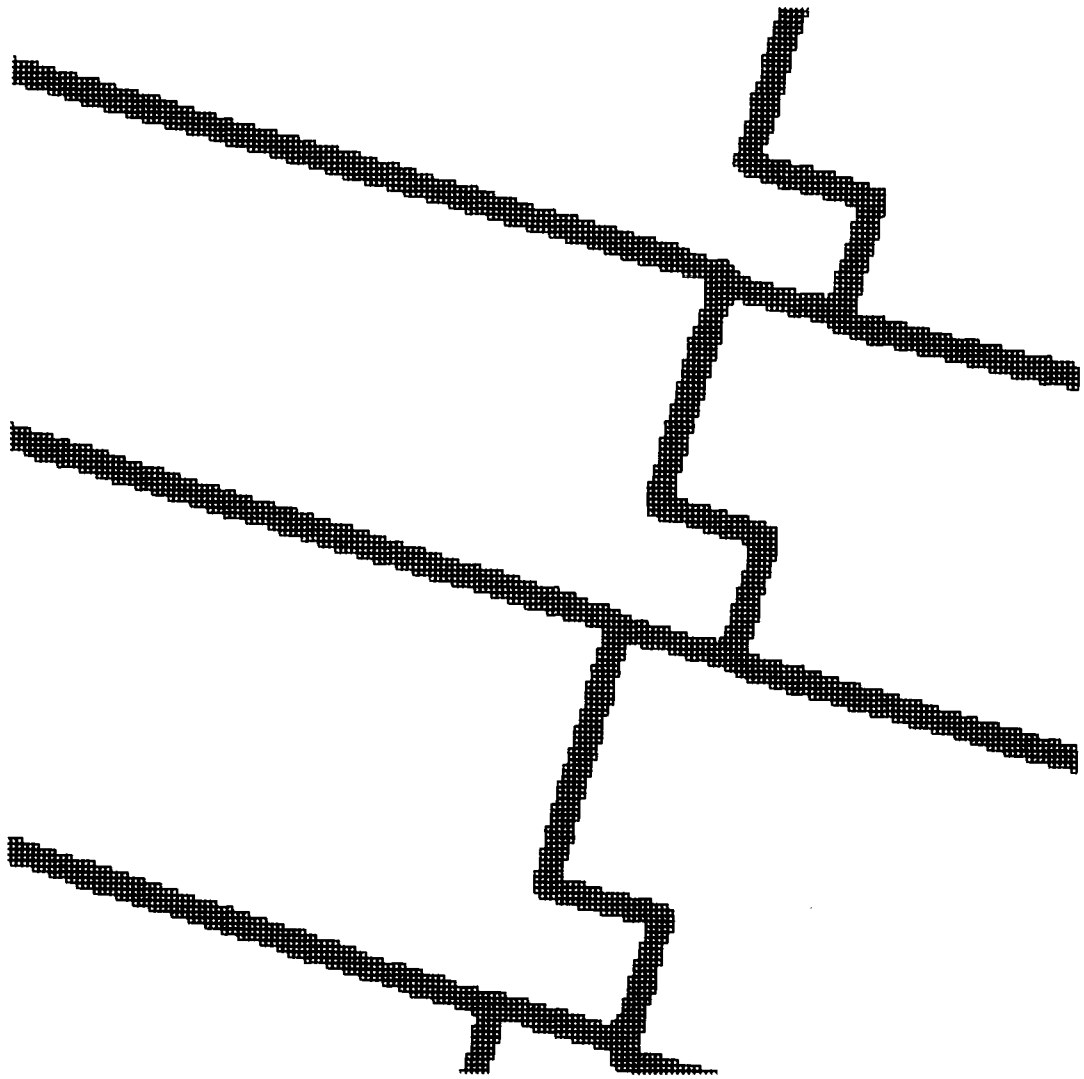


Figure 7.8c: Grossly enlarged raster image generated by Scitex scanner showing pixels describing OS linework. Crown Copyright Reserved.

The scanning process on the Scitex system was quoted as being c. fifteen minutes set-up time and a further seventy-two minutes scanning time. For the same resolution as used by Kongsberg (0.1mm), set-up time would be unchanged but scanning time would be of the order of thirty minutes on the Scitex device. No modifications were necessary to the source document, nor were there any post hoc editing and data validation stages performed. No use was made of the colour capabilities of the scanner in this application although it is expected to aid interactive feature-coding and validation procedures on a Scitex edit-station: such colour detection would be useful in scanning 1:10000 scale OS sheets for areas with no 1:1250 or 1:2500 cover, since these include black line detail and brown contours.

7.4.1.1 Comparisons:

There are several problems inherent in any attempts to compare quantitatively the raster data sets. Complete map coverage in raster form is only available with the Kongsberg and Scitex data; those furnished by Bell Laboratories were restricted to a quarter of the map sheet.

Specifically, the obvious form of digital comparison is to overlay one data set with another and examine the resulting differences and similarities. This was attempted for the

Bell, Kongsberg and Scitex data although the exercise proved unsuccessful for a number of technical and conceptual reasons. The basic problems and constraints of such an approach to comparison are itemised below.

1. The sheer volume of data comprising multi-raster images dictated that only selected sections of the map images could be held in the computer simultaneously. For this reason, any comparisons attempted were restricted to images of 500 x 500 pixels; this generated a half-megabyte storage requirement for two raster images to be processed at-a-time.
2. Since the pixels in each raster data set were of differing resolutions, it was necessary to perform appropriate scale changes to reduce them all to a common size for comparison. Since the work was carried out on the NUMAC IBM 370/168 serial processor, this was known to be a lengthy operation (see section 6.4.1).
3. The coding and compaction conventions adopted by each raster data set were different and hence software was necessary to reorganise them all to a common structure. For ease of programming and subsequent display of the results, this structure comprised the complete uncompressed pixels held scan-line by scan-line.

4. Finally, since the coordinate conventions adopted by each of the source scanners were different, it was necessary to perform translations of the raster arrays in the x and y directions and rotate them as necessary to determine a common reference system.

A number of attempts were made to perform data overlaying digitally by these techniques although no useful results were achieved. It proved most difficult to perform the translations and rotations since these had to be determined empirically on a trial-and-error basis. The lack of appropriate raster-based graphics facilities in Durham at that time (see section 5.3.5) meant that any assistance of graphics (which would have been invaluable) were also lacking. A number of reasons were concluded from this inability to perform a working digital overlay system. For instance, the efficacy of performing a scale change for comparison purposes under these circumstances is not considered completely legitimate; a scale increase suggests attempts to extract more information than is actually held in the original data. More appropriately, though, the orders of this scale change should probably have minimal importance. The results achieved are partially a reflection of the quality and the characteristics of the transformation routines (which were modified from analytical photogrammetry

operations known by the author) and of the reliability of the identified fiducial marks (all of which were, by necessity, OS map features with lines of several pixels wide). The alterations of the coordinate systems used from those described in the technical specifications and (in some cases) the non-square pixels in the raster images were also suspected reasons for the failure.

For similar reasons, the use of photographic techniques for data overlay - although considered - were also rejected since the photographic process would have been affected by similar problems as those encountered by the digital process. The pixel data in the Bell Laboratories' data, for example, were known not to be square and hence it was considered unrealistic to continue the development of effective digital or photographic comparisons, given the drawbacks encountered after about one month's work.

Ultimately, then, any valid comparisons must rely on a visual inspection of the resultant graphics (figures 7.6a,b,c, 7.7, 7.8a,b,c). At the outset, the Scitex images appear to have most potential for precise large scale work; the image is clear and accurate, with no visual signs of noise or coding errors. The selected scanning resolution is adequate for image representation with no visual effects of stepping at 1:1250 scale. The Kongsberg data cannot be

compared fairly by graphical means with the example shown in figure 7.7 since this image comprises inappropriate thresholding. Nevertheless, from a series of modified images displayed on a graphics display terminal, it is clear that substantial amounts of noise is present in the image. The resolution (ten lines per millimetre) chosen by Kongsberg on the basis of their own experience is half that chosen by USGS and stepping is visible on 1:1250 scale output. Clearly, it is advisable to scan 1:1250 documents with a resolution of twenty lines per millimetre; scale reductions are then possible post hoc (if required) for other processes. Scanning times for the KartoScan to encode the data at twenty points per millimetre are expected to increase by a factor of four. This suggests a scanning time of c. 40-60 minutes (cf. the Scitex machine has taken 72 minutes).

In general, the Bell Laboratories' scanner cannot be compared fairly with the equipment from Scitex and Kongsberg; the scanner costs one sixteenth of its more sophisticated competitors. However, the raster results from Bell are generally good within certain lower thresholds - the system presents value for money for appropriate tasks.

In conclusion, it should be pointed-out that all the scanners tested are sufficiently advanced to handle Ordnance Survey data. Their design and technical specifications are

capable of accepting existing documents and scanning them with sufficient speed and pixel resolution. The levels of noise inherent in the resultant digital data are tolerable, providing device calibration and threshold values are accurately established at the set-up stage. However, the cost-effectiveness of each system cannot be completely assessed without further, more detailed, appraisals and a specification of Ordnance Survey work methods. It may, for example, be uneconomic to reduce or enlarge photographically the source documents if some other scanners (at the same price) will accept full-sized conventional MSD's. It has proved impossible to investigate the effect of poor-quality source documents because of the limited available access to scanners during the research project: it has already been pointed out that these are expected to be a major constraint on automated data capture generally and will slow-down progress to a presently unknown extent.

The ability to convert the raster-files directly into vector notation is likely to be of primary importance, at least for the next ten years. It is the performance of this conversion process which will have most importance for Ordnance Survey applications if vector storage remains the favoured mode. All the systems tested provide post hoc software options to generate vector data; these are described and compared in section 7.4.2. No detailed account can be

given of the design of the algorithms as, in most instances, these are company secrets. For these reasons, comparisons below are carried out on an empirical basis. However, there is no reason why scanning hardware and vectorising software from different suppliers cannot be interchanged to provide customised installations although the wide diversity of data formats and computing hardware suggests this approach to be arduous or, at least, demand interfacing problems.

7.4.2 The characteristics of the raster-to-vector conversions

At the outset, it is worth pointing out that the experiments carried out in this research have primarily (almost exclusively) addressed the problems of encoding the geometry of the line work, not the feature attributes. The latter task is likely to involve substantial access to appropriate interactive systems which were not available to the author. For these reasons, most emphasis is placed on the conversion of the pixel data to vectors and the likely loss of quality in the vector files over the raster images and (perhaps more importantly) the original scanned documents. The problems associated with an effective raster-to-vector conversion system, generally, have been discussed widely elsewhere (Harris et al 1981, Loodts 1981, Peuquet 1981, Weber 1981). The characteristics of the

resultant vector strings are not amenable to most digital manipulation other than data compression and somewhat poor-quality graphics: there are no conversion systems available at present to suit the quality and data structure requirements of the Ordnance Survey. Specifically, any vectors generated automatically have specific limitations at present:

1. The data structures are inappropriate for most tasks other than graphics displays; even systems generating link and node files do not match the level of information and the cleanliness achievable with manual digitising systems (e.g. PMA 1978).
2. The organisation and the ordering of the vector segments resembles the scan-line by scan-line properties of the original raster-file; linear features are seen to be terminated once they exit from the current strip of c. 10-20 scan-lines (see section 5.3.1).
3. In particular, most vectorising software has difficulty in defining node points (line junctions) accurately. In extreme cases, they are replaced by linear structures comprising several short links in the general position of the node. Some examples of this problem are evident in several instances in the figures showing the results of

the raster-to-vector conversions. In particular, see figures 7.9b, 7.11b,c 7.17 ad 7.18 where elongated nodes abound.

4. Vector spiking is a characteristic problem associated with existing raster-to-vector methods; it occurs in the skeletonising process of lines with varying pixel-widths. The result is a series of very short 'spikes', perpendicular to the general direction of the feature. A number of research projects are attempting to resolve this problem - in particular, the Oxford Image Analysis team (Preston pers comm 1981) and that at IGN, Brussels (Loodts pers comm 1981).

5. The volumes of data involved and the necessarily serial-nature of the conversion algorithm is causing severe timing difficulties on all types of computer (serial and parallel).

In general, and despite the above comments, the results attained from the tested serial-systems are encouraging although none offer complete solutions for Ordnance Survey work methods. The tests undertaken with the Bell Laboratories' system, although interesting, show results which are unacceptable for most cartographic processes.

ORDNANCE SURVEY

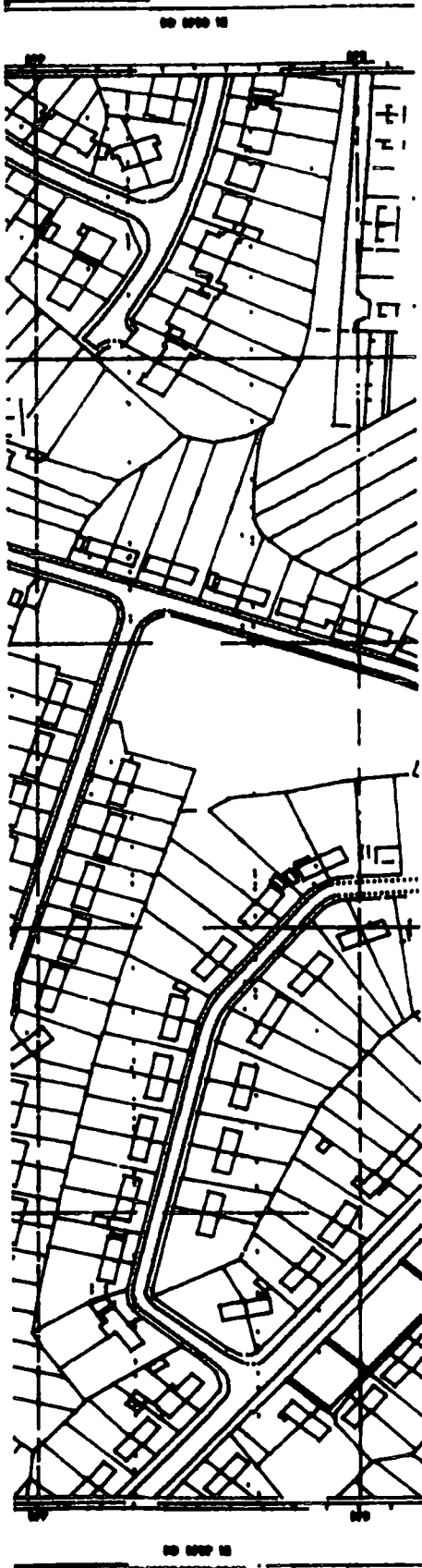


Figure 7.9a: Strip of OS data in vector form generated by Bell Labs. conversion software with Bell raster data. Crown Copyright Reserved.

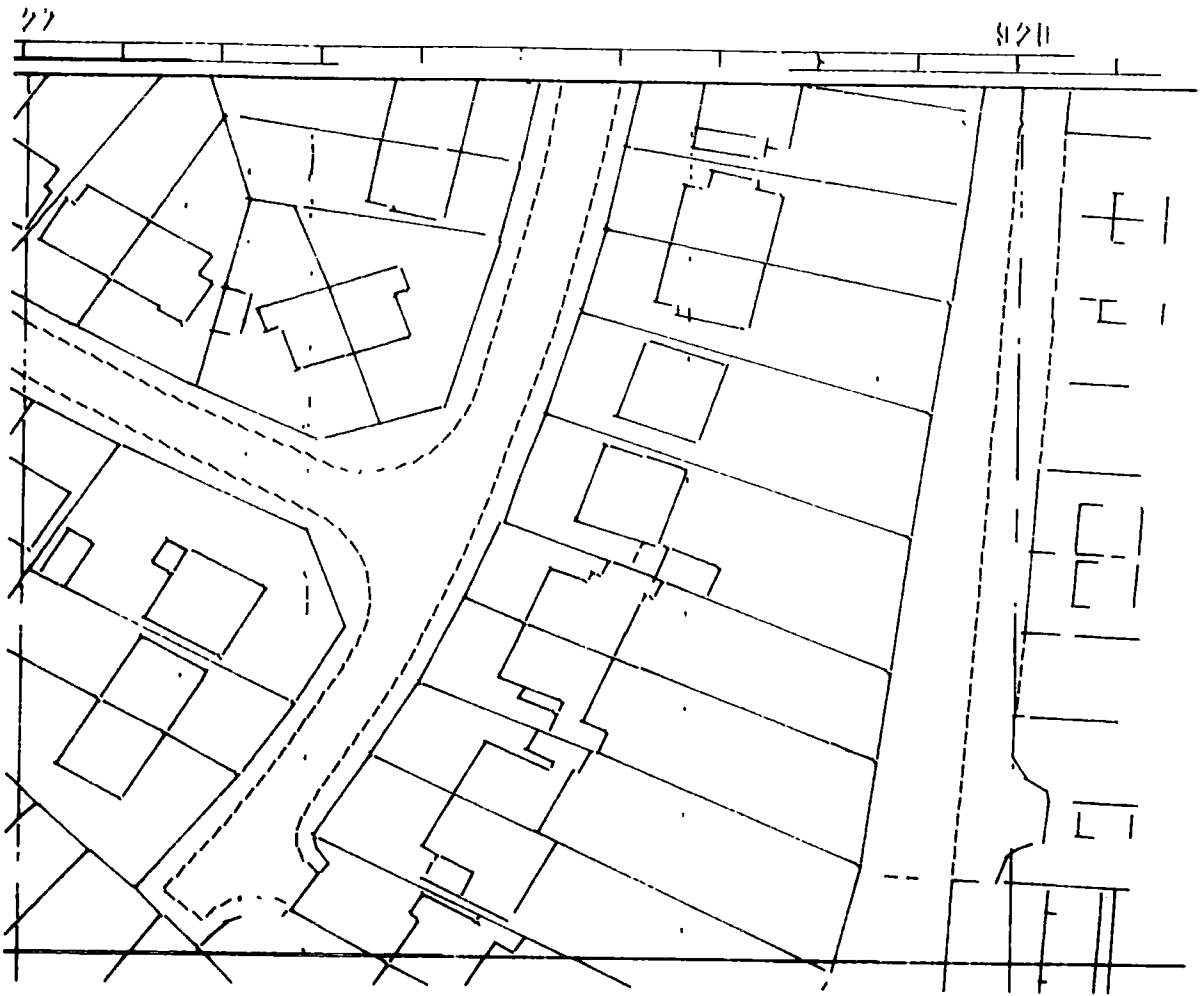


Figure 7.9b: Single grid-square of OS data in vector form generated by Bell Labs. software with Bell raster data. Crown Copyright Reserved.

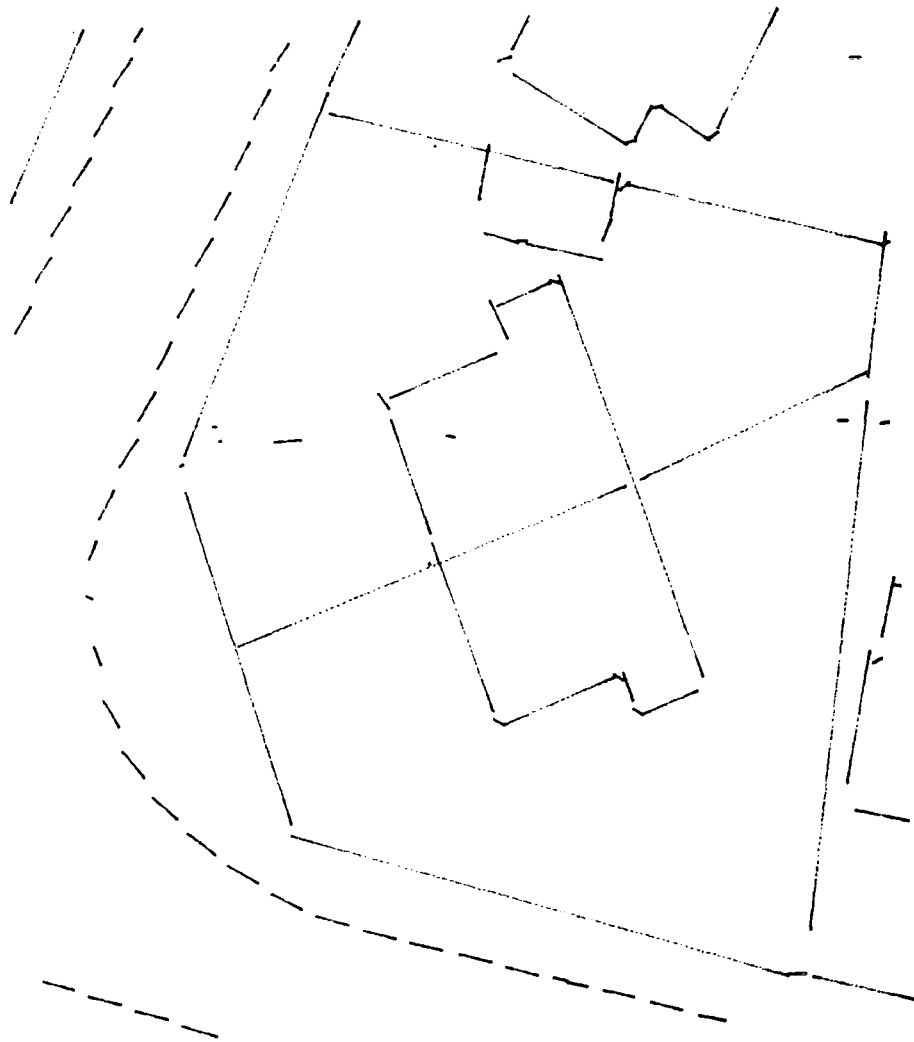


Figure 7.9c: Two land parcels in
vector form generated by Bell Labs.
software with Bell raster data.
Crown Copyright Reserved.

Although the results from their EDAD scanner are sufficiently accurate if materials are photo-enlarged to give 0.05mm resolution, the criteria for the production of vector data are not - these have been pointed-out in 7.3.2 and are described in detail by Pferd and White (1981). Figure 7.9a,b shows the results obtained. The (somewhat) poor appearance of the vectors is to be expected from a system which is primarily not designed for high-quality cartographic work. Figure 7.15 shows the same data plotted on a high-quality Ferranti Master plotter at the Ordnance Survey headquarters. A software interface was written to plot all the vector files on this plotter to ensure that the quality of the results are not effected by poor vector plotting facilities.

It can be seen how the Bell software has an apparent desire to rotate all vectors through a small, constant angle; this is especially clear with the pecked-line data and buildings which have lost their 'square' properties. The poor-quality of line junctions (node points) are unacceptable for even the crudest map displays; in most cases, there are intolerable gaps at each junction and the introduction of several short vectors at each node has distorted the geometry of the data. The errors at each node point are not generally consistent; some lines overshoot each other whilst others terminate too soon creating gaps. The software has particular problems in handling the junctions of lines

converging at a small angle. In these instances, each node point is seen to be replaced by a short-length vector. It is likely that these deficiencies are largely a result of inappropriate tuning of point filtering tolerances at the chaining and smoothing stages since the Bell system is designed to convert to vectors quickly (with low order tolerances) to produce an approximation vector image for rapid transmission of the digital image by telephone between sites.

The effects of scanning and vectorising textual data are shown clearly in figures 7.9a and 7.15. The map title and grid-values have appeared in the raster file (see figure 7.6a) since a small area outside the neat-lines was scanned in addition to the main map-detail. The resultant vector representations of this text is indecipherable, which is surprising since most engineering drawings comprise text information which is preserved in the Bell digital system.

In general, the Bell Laboratories' vectors are efficient in plotting speed - largely a result of the requirements of a teleconferencing system. The data are 48.8 per cent efficient in plotting (i.e. 48.8 per cent of the plotter pen-movement is in the down position). Table 7.10 shows a summary of the vector characteristics although the Bell data are (by necessity) for one-quarter only of the total map

Table 7.10: A summary of the characteristics of the vector files produced by the tested commercial systems

(a) Data produced by conventional OS digital flowline:
(source Adams 1979).

Total line length = 4369.6 cm
Number of points = 13687
Number of segments = 11857

Max. inter-point distance = 4.3 cm
Min. inter-point distance = 0.0 cm

(b)	Bell Labs EDAD	Kongsberg SysScan	Oxford PEPR	Oxford Scitex	USGS Scitex
	-----	-----	-----	-----	-----
No. of segments	3635	12165	15946	9385	4102
No. of points	7267	30056	34899	23380	11669
Max pts/segment	2	15	18	30	20
Min pts/segment	2	2	2	2	2
Av. pts/segment	2.0	2.5	2.2	2.5	2.8
Total pen-down line length	1111.4	4386.2	3444.7	3946.6	4417.6
Plot length as % of vector file	101.7	100.4	78.8	90.3	101.1
Max inter-pt dist	38.8	5.2	4.8	7.8	14.1
Min inter-pt dist	0.0	0.0	0.0	0.0	0.0
Av. inter-pt dist	15.3	0.4	0.2	0.4	1.1
Total pen-up line length	1167.2	14608.5	202376.5	108258.9	5249.9
Plot efficiency	48.8	23.0	1.7	3.5	45.7
Plot time on Ferranti Master	0h 40m	2h 30m	7h 00m	5h 25m	1h 55m

All distances are in centimetres; plot efficiency is percent

image (section 7.4.1). Each vector segment comprises two end-points - this is the convention adopted in the Bell vector data structure which is summarised in appendix 7.1. The large average length of the vector segments (15.3 cms) reflects the criterion of the Bell system - to give a general impression only; most of the cartographic systems (table 7.10) restricts vector segments to much smaller quantities (typically 0.4 cms).

The vector data resulting from the Kongsberg SysScan are shown in figure 7.11a,b,c with an example of the output from the Ferranti Master plotter in figure 7.16. These results show the extent of the limitations with most serial-based vector conversion systems which were pointed-out earlier. It is the precise geometry of the large scales data which demonstrates most clearly the errors in the vectorising process; such inconsistencies are difficult to detect in data with less geometric properties (e.g. contours). A significant problem in the Kongsberg results has seemed to arise with the edge-matching of data across each swathe in the full raster image (section 7.3.3); most features crossing to adjacent swathes appear to deviate marginally along the swathe edge. This causes a series of linear distortions appearing as parallel lines running north to south along the complete graphic. Other problems have arisen with the square nature of buildings which is lost to a large extent in the

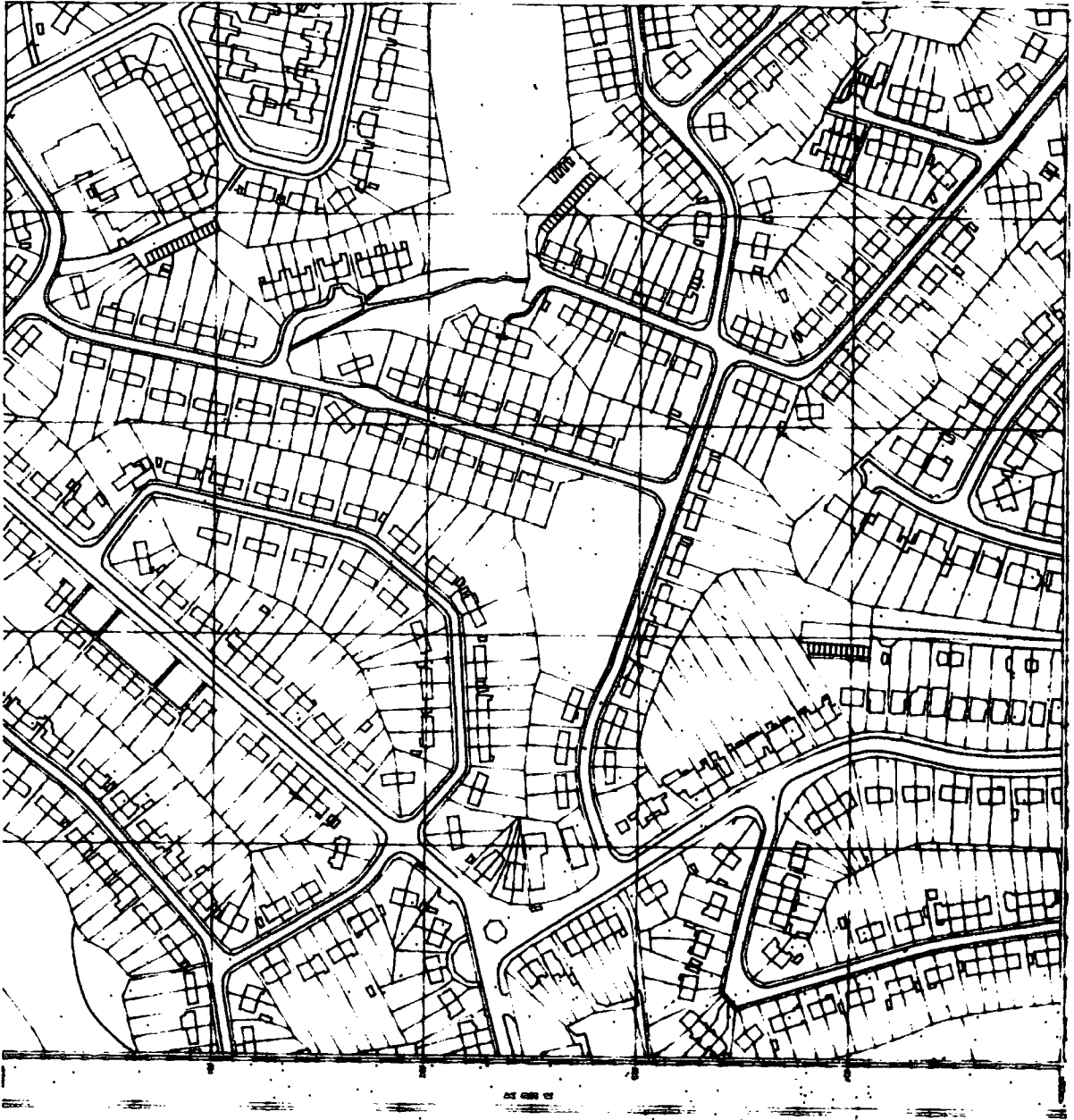


Figure 7.11a: Complete OS map in vector form generated by Kongsberg conversion software with Kongsberg raster data. Crown Copyright Reserved

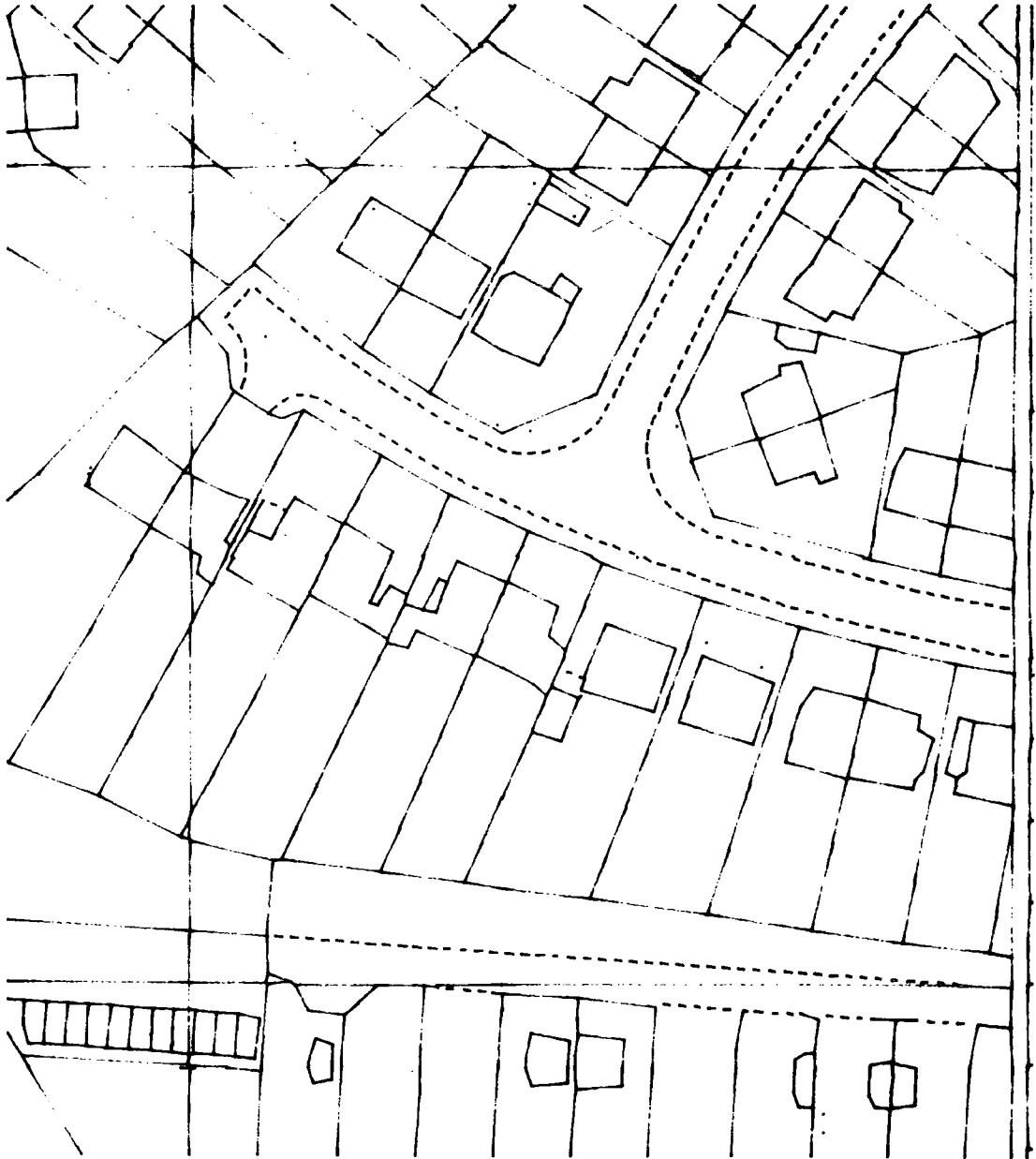


Figure 7.11b: Single grid-square of OS data in vector form generated by Kongsberg software with Kongsberg raster data. Crown Copyright Reserved

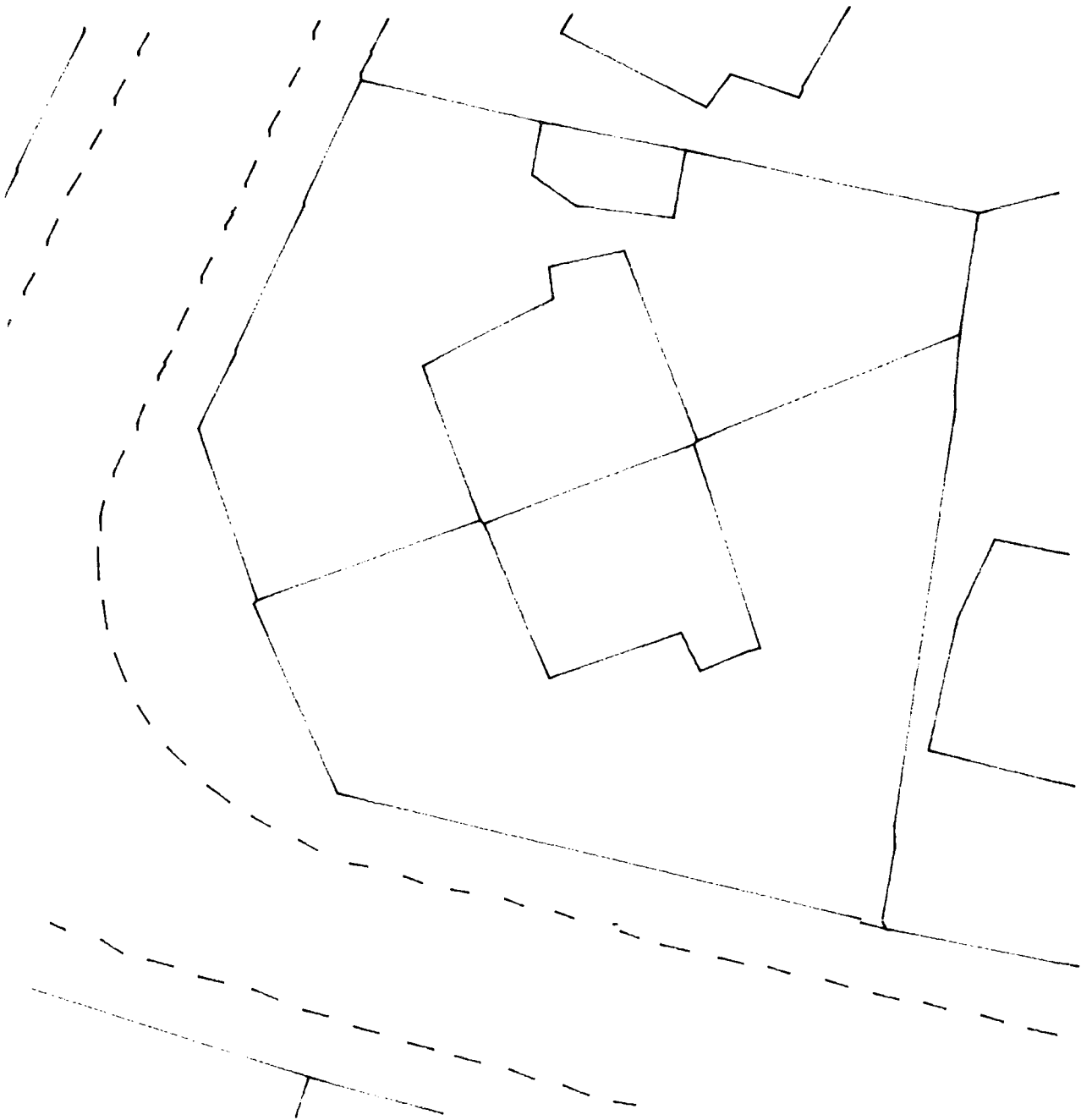


Figure 7.11c: Two land parcels in vector form generated by Kongsberg software with Kongsberg raster data.
Crown Copyright Reserved.

Kongsberg results (cf. Bell Laboratories); though there is little evidence of rounded-corners, each supposedly 'square' building comprises more than four points resulting in a distorted image. Building squareness is likely to be dependent on the point filtering/line smoothing stage. The graphic evidence suggests the need for a post hoc data cleaning stage to make-square the building data though this can only be done efficiently after the assignation of feature codes describing the polygon as a building; proven methods for this task are already available in the existing Ordnance Survey digital flowline (section 2.5). Fences and property boundaries appear jagged and often contain several gaps along their length, probably for similar reasons. This problem is shown even more clearly in the jagged grid-lines.

In general, the Kongsberg system handles the pecked-line data very well; they have been reconstructed consistently with inter-peck spacing and line length agreeing with the data appearing on the original graphic. This is surprising since the close proximity of line-work has generally caused serious problems in the accurate reconstruction of other vectors by the Kongsberg software: there is evidence of extremely small polygons being generated at some line junctions, in others, complete geometry is lost; there are several instances of line undercut at node points but no

evidence of overshoot as was the case with the Bell Laboratories' vectorisation. In some instances, the nearness of close parallel lines (see figure 7.16) has caused chaos in the vector reconstruction; up to fifteen small links have been inserted between the relatively short parallel lines, which has attempted to concatenate them into a single, thicker line. The Kongsberg GINIS software may cope with these errors although, generally, most data structures will require software tuning or post hoc validation of data to resolve them first. Generally, the results from the Kongsberg system have provided good examples of the limitations cited earlier in present raster-to-vector strategies. In short, on the basis of these tests the geometry and structure of the data produced is unacceptable in its present state for most non-graphic manipulation and has severe constraints even for graphic presentation (though the quality of data scanned at 0.05mm resolution may be much better).

The total line-length of the vector data is very similar in both the Kongsberg and Scitex results - these agree within thirty centimetres of each other (table 7.10). Indeed, the characteristics of the the Kongsberg and Scitex data are similar to those describing the vector file originating from the Ordnance Survey digital flowline (table 7.10). The Kongsberg results agree within seventeen centimetres of the

currently defined 'true' length although point-frequency is very much higher with the automated method. The larger number of x,y points in the Kongsberg file (c. 2.2 times as many as OS flowline) gives further impetus towards the need for more appropriate software tuning and point-filtering.

The raster-to-vector software originating from Oxford University is one of the few systems known to the author which generates data in a link and node format, thus providing more topology and easier interfaces for other software (section 5.6.2, Harris et al 1981). Thompson (1978) pointed out that link and node notation is expected to be the norm in future Ordnance Survey applications. For these reasons, two investigations into the quality of the Oxford algorithm have been undertaken: these used both the raster data originating from the Oxford PEPR scanner (7.3.4) and the higher-quality raster data generated on the more sophisticated Scitex scanner at the USGS (7.3.5). Figure 7.12a,b,c shows the resultant vectors generated from the PEPR raster data and, for comparisons, figure 7.13a,b,c shows the results from the same software using the Scitex data. Marked differences are visible. These examples show conclusively how the quality of resultant vectors is not only dependent on the design of the conversion algorithm but also on the quality of the raster data used as input. Without access to the Oxford University data base retrieval software, it has



Figure 7.12a: Complete OS map in
vector form generated by Oxford Univ.
software with Oxford Univ. PEPR
raster data. Crown Copyright Reserved



Figure 7.12b: Single grid-square of OS data in vector form generated by Oxford Univ. software with PEPR raster data. Crown Copyright Reserved

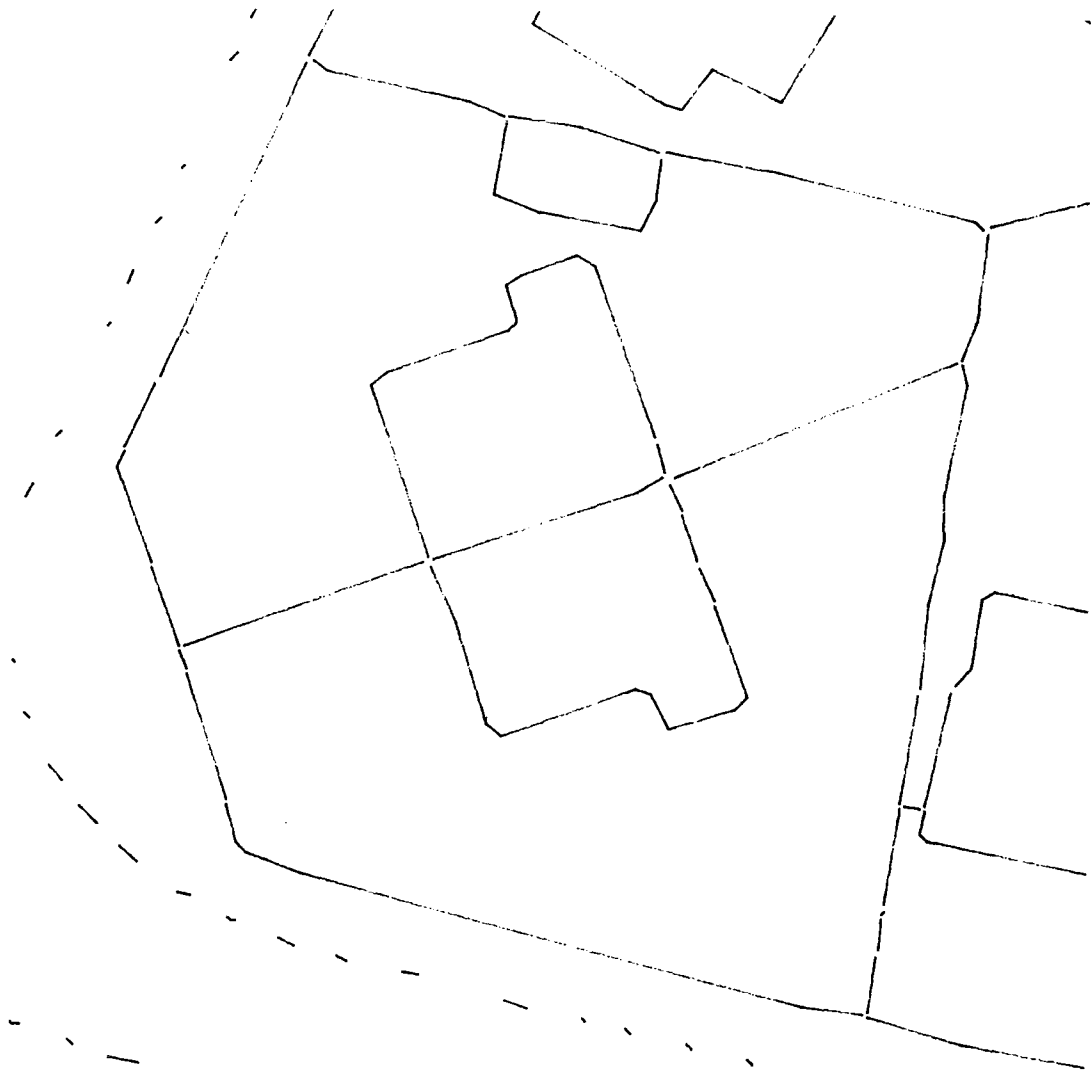


Figure 7.12c: Two land parcels in vector form generated by Oxford Univ. software with Oxford PEPR raster data. Crown Copyright Reserved.



Figure 7.13a: Complete OS map in
vector form generated by Oxford Univ.
software with USGS Scitex raster data
Crown Copyright Reserved.

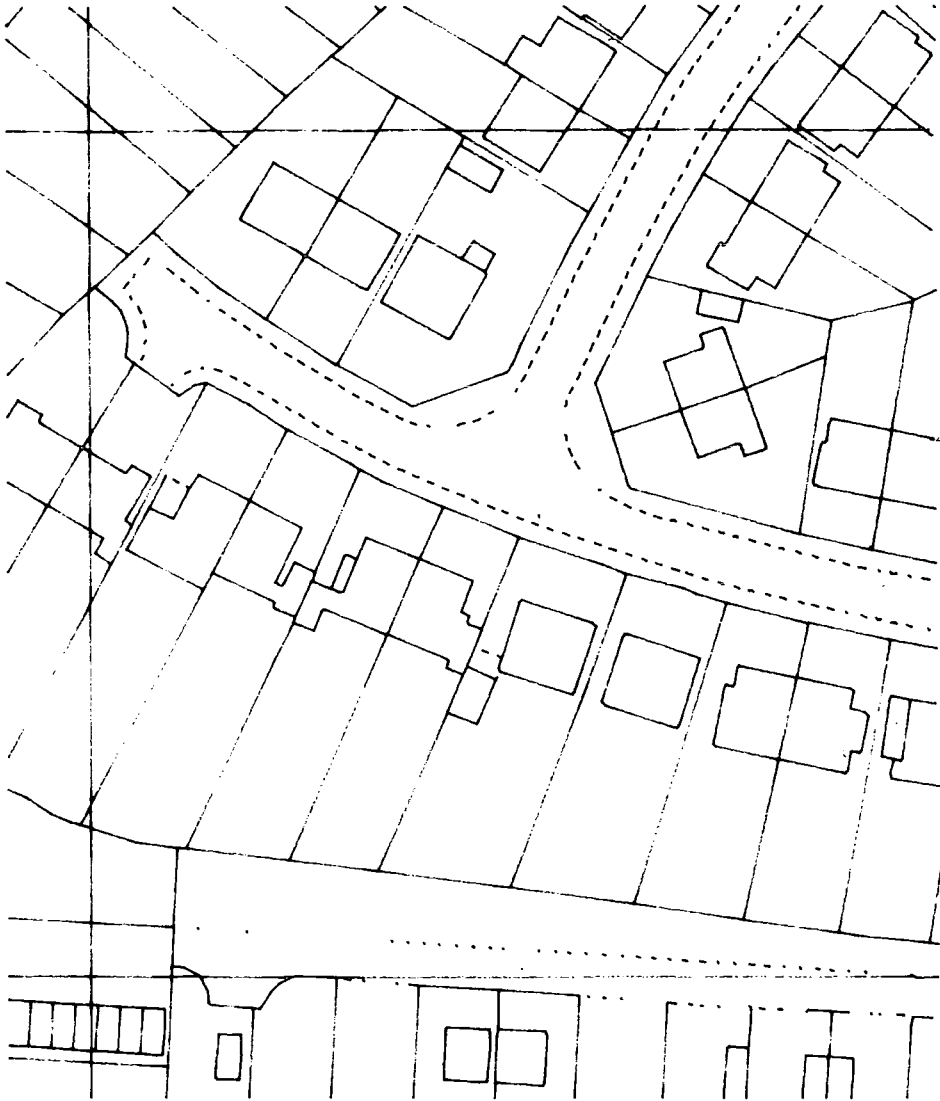


Figure 7.13b: Single grid-square of OS data in vector form generated by Oxford Univ. software with Scitex raster data. Crown Copyright Reserved

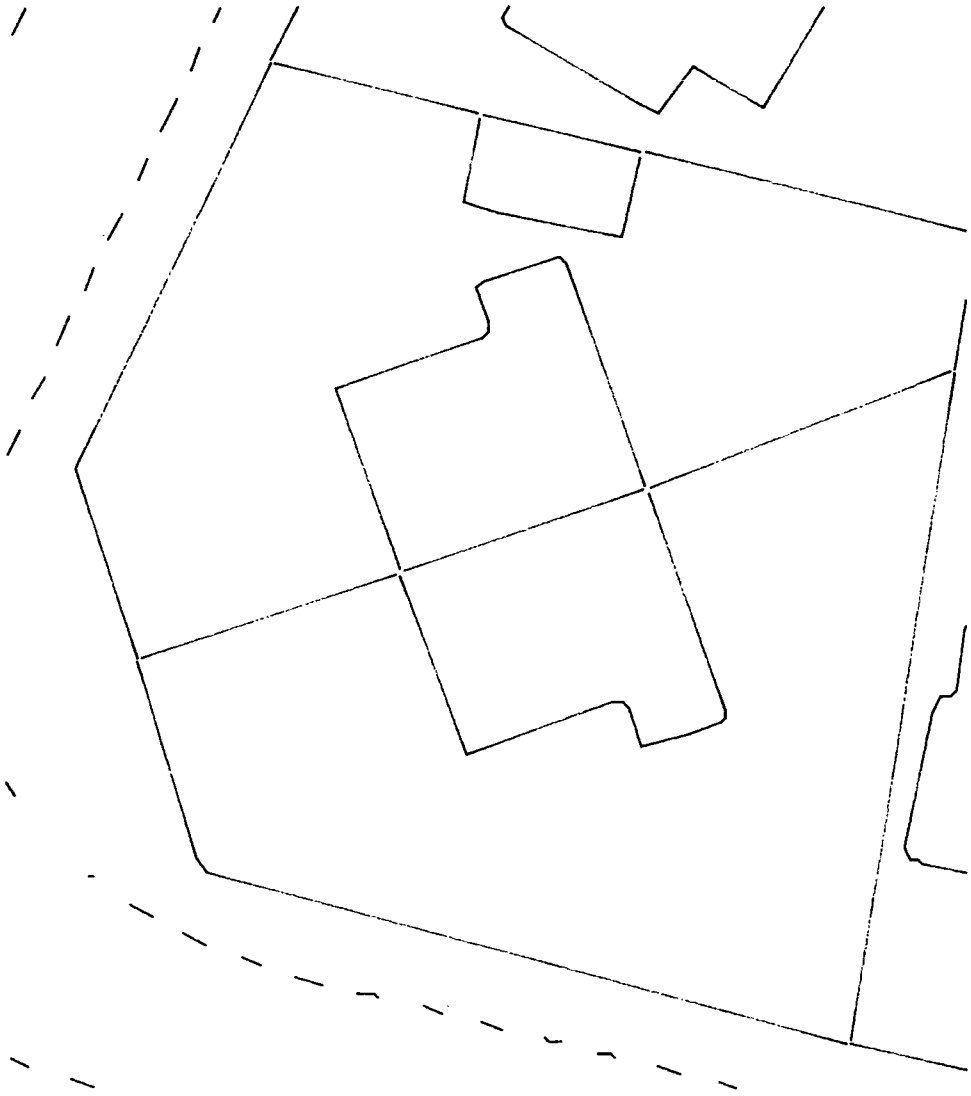


Figure 7.13c: Two land parcels in vector form generated by Oxford Univ. software with USGS Scitex raster data. Crown Copyright Reserved.

been difficult to cross-reference the link and node files to generate specific polygon data. For these reasons, only the links files have been plotted in figures 7.12 and 7.13, causing gaps to be inserted at each node position. These same reasons account for the poor plotting efficiencies described in table 7.10 for the Oxford data; much improved plotting times are possible through recourse to sensible link and node connection by appropriate software. The results shown in figures 7.17 and 7.18 give an indication of the level of quality possible on the more sophisticated Ordnance Survey plotting devices; these were drawn on film by the Ferranti Master plotter for the PEPR and Scitex data respectively.

In general, the results obtained from the PEPR scanner are surprisingly good, especially since a ten-times photographic reduction of the source document is necessary prior to scanning (section 7.3.4). However, a closer examination of the results with a linen-tester or by increasing the plotting-scale (figure 7.12c) shows some grave deficiencies more clearly in the PEPR results. Reasons for these are difficult to identify and a number of factors are believed to be responsible - a comparatively poor scanner, the effects of photo-reduction and the post hoc vectorising and filtering. The Oxford team have suggested most faults to lie with the PEPR scanner which has an architecture and signal-to-noise

ratio which is inappropriate for cartographic work (Preston pers comm 1981).

Although the Oxford-vectorised data reflect most closely the storage notation of the Ordnance Survey restructured data (section 2.7), they differ markedly from those attained by manual digitising (table 7.10). There are c. 2.5 times as many points in the Oxford files with a reduced overall line-length. To some (unknown) extent, the large number of coordinate points is largely a result of the high-frequency of pecked-lines on the source document. In existing OS conventions, pecks are not held explicitly in the data archive and are generated post hoc by software. In general, the Oxford algorithm has extreme difficulty in handling pecked data; this is evident in both the results from the PEPR and Scitex scanners (figures 7.12 and 7.13). This is a major deficiency from an Ordnance Survey viewpoint since about twenty per cent of line-work is pecked with present map specifications. The removal of the node-points in the graphic displays (figures 7.12 and 7.17) has enabled inconsistencies in the geometry of the PEPR data to be seen more clearly. A large proportion of the gaps in the features could be removed without serious loss of detail. This increase in link and node references has provided a measure of the incompleteness of the data geometry from the PEPR; more sound results are evident with the Oxford/Scitex results

although there are still one or two errors present (figures 7.13 and 7.18). This over-production of node-points has caused many buildings to lose their squareness and other features to be distorted generally.

On the whole, the Oxford raster-to-vector system has been more encouraging when presented with higher-quality raster data at the input stage. A much neater data structure has been generated with the Scitex raster file (table 7.10): there are fewer links than with the PEPR data producing a much more concise, less-jagged series of lines. However, there remains a high-frequency of very short links - especially at node positions. Point smoothing is a little too coarse for curved features, making them lose their true characteristics; a roundabout symbol, for example, is reduced to a sixteen-sided polygon and most building corners are rounded. These errors are largely due to inappropriate software tuning, which also accounts for the inability to reconstitute the pecked-lines. Even with the Scitex data, however, pecks have not been reconstructed successfully by the then existing Oxford software (cf. Kongsberg). Most have correct orientations but have various lengths ranging from dots to marginally longer pecks than on the original foil. As with the Kongsberg results, it can be concluded that much data validation and cleaning is necessary before they can be presented to a data base system for archiving.



Figure 7.14a: Single grid-square of OS data in vector form generated by USGS software with Scitex raster data. Crown Copyright Reserved.

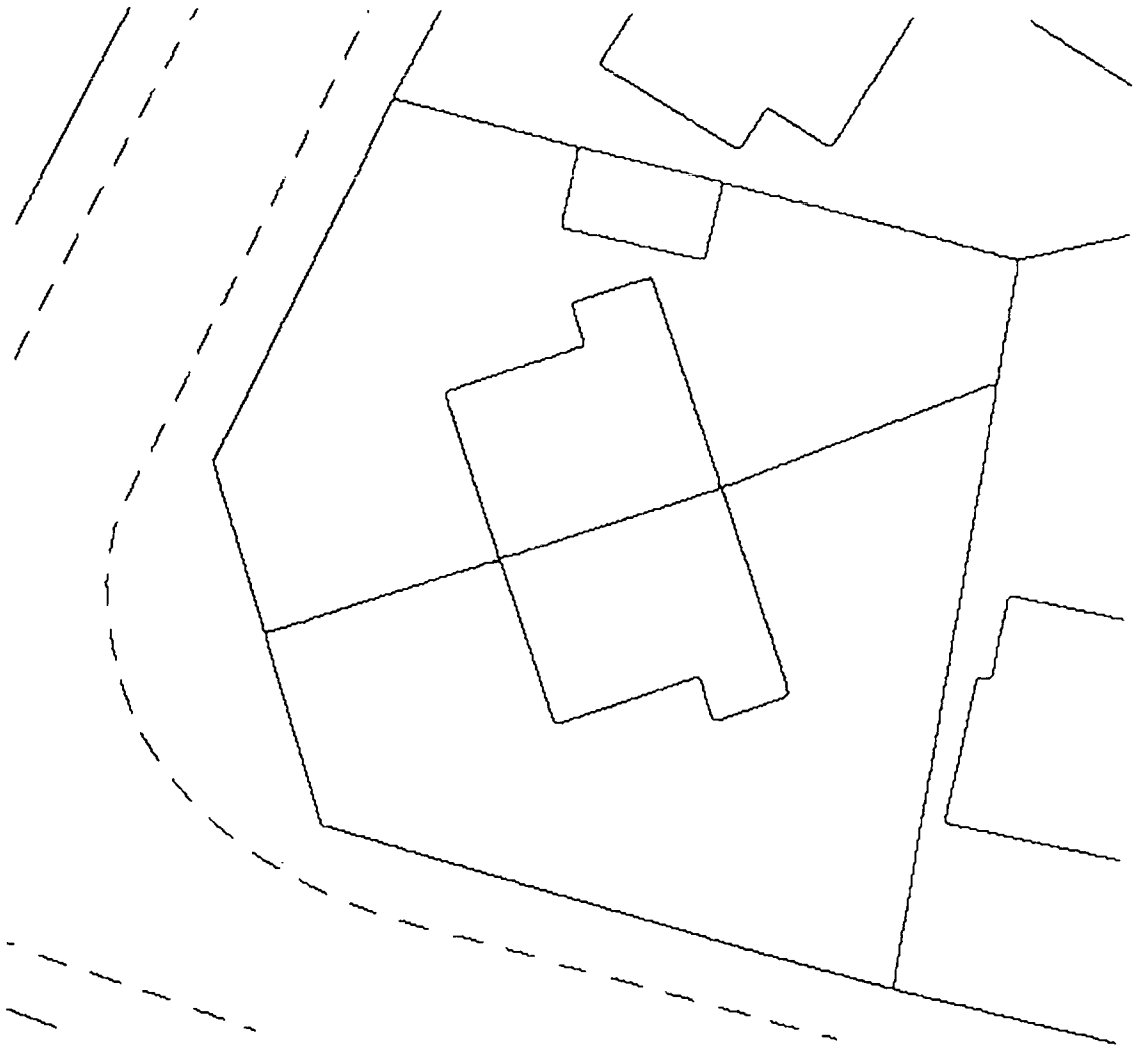


Figure 7.14b: Two land parcels in vector form generated by USGS software with Scitex raster data. Crown Copyright Reserved.

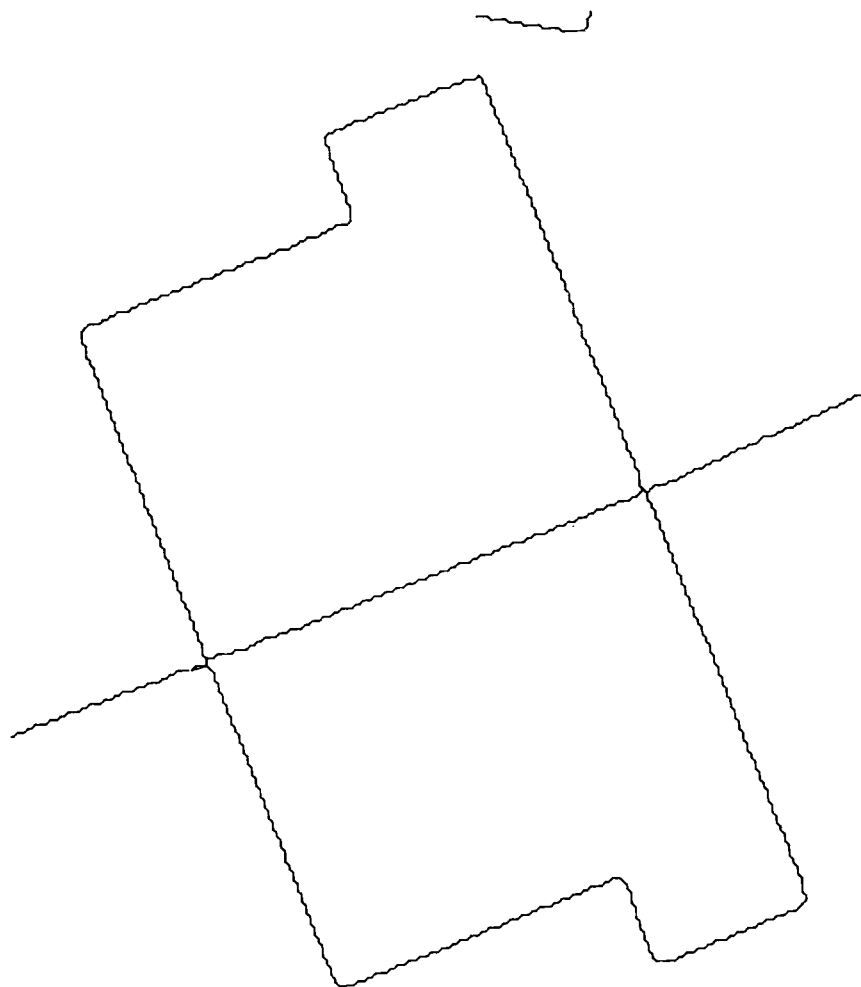


Figure 7.14c: Semi-detached property
in vector form generated by USGS
software with Scitex raster data.
Crown Copyright Reserved.

The quality and accuracy of the vectors emanating from the USGS/Scitex system meet the standards required by the Ordnance Survey more closely although there is little topology inherent in their organisation. Examples of these results are shown in figure 7.14a,b,c; the image produced on the Ferranti Master plotter is at figure 7.19. The software installed at the USGS was developed by the Computer Science Division, NRIMS, CSIR, Pretoria, South Africa under the direction of Dr. Ray Boyle (Boyle pers comm 1981) and is described by Boyle (1979). The data furnished by the USGS have been skeletonised and pixel-chained only, without any smoothing or interactive editing. This has enabled an estimation of the percentage success of the raster-to-vector conversion prior to any point-filtering operations; it is likely that many of the problems of the Kongsberg results, for example, are not a reflection of the conversion to vectors, rather the inappropriate tuning of the point-filter after vectorisation (see earlier in this section). A plot of the vector data converted from raster prior to smoothing produces line-work which is somewhat jagged; this is virtually indistinguishable at 1:1250 scale but is visible through a linen-tester or in enlarged plotting scales (figure 7.14c). The jagged appearance is removed when the data are passed through an appropriately tuned point-filter; this improves the appearance of the graphic-displays, speeds-up plotting times and removes redundant data.

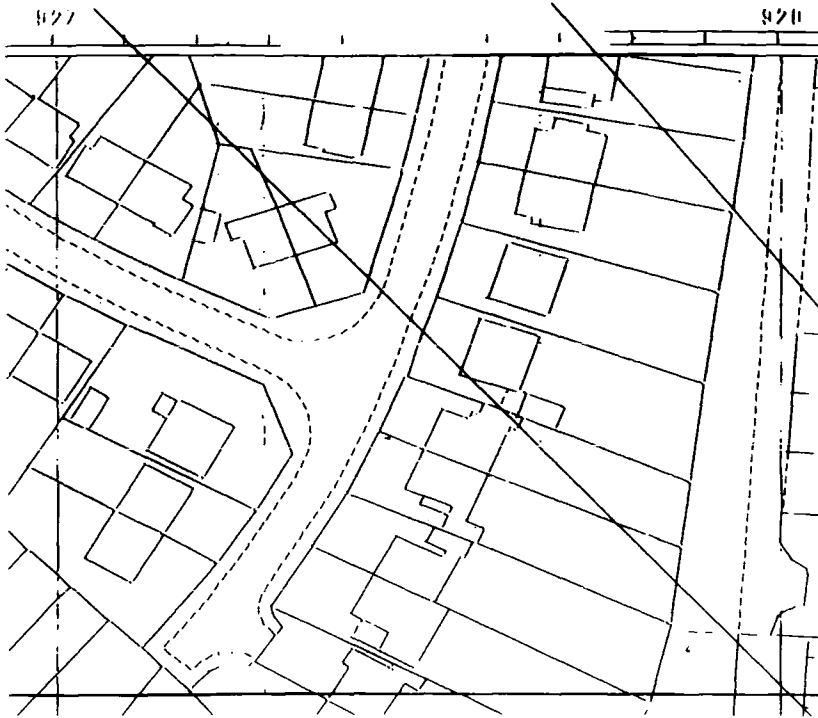


Figure 7.15: Bell Labs. vector data

Figure 7.16: Kongsberg vector data

(both produced on Ferranti Master
plotter). Crown Copyright Reserved.





Figure 7.17: Oxford PEPR vector data

Figure 7.18: Oxford Scitex vector data

(both produced on Ferranti Master
plotter). Crown Copyright Reserved.

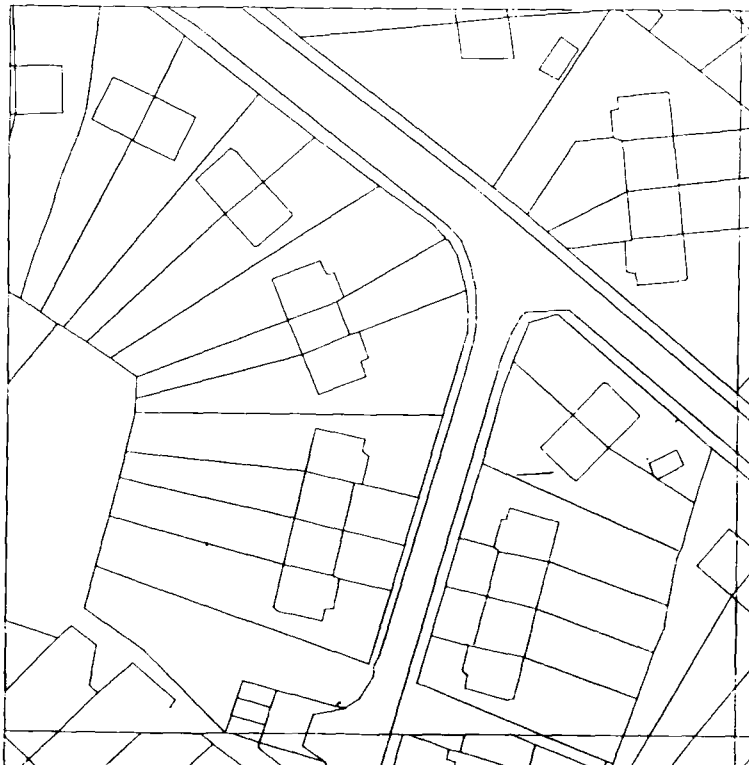




Figure 7.19: Unfiltered Scitex vectors

Figure 7.20: Filtered Scitex vectors

(both plotted on Ferranti Master
plotter). Crown Copyright Reserved.



Approximately ninety-eight per cent of the points are removed by line-smoothing. The results are shown in figure 7.20 for the USGS vectors. Prior to smoothing, the data required c. ten hours to plot on the Ferranti Master plotter; this has been reduced to one and a half hours afterwards.

The quality of the vectors is generally good; the network of interconnected lines is neat and geometrically consistent. From a graphical viewpoint, this algorithm is most appropriate for Ordnance Survey needs although the resulting data structure and processing times remain a major drawback.

Thus, the data produced automatically by each tested system are similar in content although they differ widely in quality and format. Each system has generated a series of vectors describing the source document which was scanned. There has been no data validation or attempts to attach any feature-coding to the line-work; the problems and resources necessary to do this are untestable without lengthy access to appropriate hardware and a thorough knowledge of their uses. Clearly, it has been important to monitor the times taken by each system to reach this stage in the data entry process: an idea of processing times has direct relevance to the expected time-scales necessary to raster encode the national topographic data archive in the UK (HMSO 1979). However, a detailed breakdown of the scanning times is difficult since

any information obtained has been restricted to those figures quoted by the machine operators who performed the scanning trials. A summary of the raster-to-vector performance rates are given in table 7.21 where it can be seen that each of the systems compare favourably with each other - with the exception of the USGS which was an order of magnitude slower. Furthermore, these rates of processing appear to match the times quoted for the parallel strategies for raster-to-vector attempted on the ICL DAP which were described in chapter six. It was pointed out in section 6.4.2.1 that a raster-to-vector conversion process is, essentially, serial in nature and cannot use a parallel architecture of the DAP to full advantage. For these reasons, then, the DAP was programmed to operate in a sequential-type mode for this particular process and hence this accounts for the similar rates of throughput per unit access time of the cpu on both a parallel and serial computer. Nonetheless, the DAP method of processing has had the advantage of generating the vectors in a structure more appropriate for further processing although since only polygons are given (and not individual segments between nodes), then further breakdowns may be necessary post hoc.

Table 7.21: Summary of the processing times necessary for raster-to-vector conversion

Figures quoted by suppliers.

	Bell Labs EDAD -----	Kongsberg SysScan -----	Oxford PEPR -----	Oxford Scitex -----	USGS Scitex -----
Computer system	Multi-user PDP 11/70	Dedicated VAX 11/780	Multi-user DEC 10		Dedic. 11/60
Vectorising (cpu mins)	unknown	c. 20	44	46	235
Smoothing (cpu mins)	unknown	unknown	1.6	3.3	5.2

In terms of run costs, the Oxford system must be the cheapest since this software has performed the conversion most rapidly. Though the Kongsberg SysScan system has performed the conversion in half that time, this is primarily due to the lower resolution pixels created: section 5.5 has pointed-out that increased pixel resolution has severe effects on processing rates. The high quality of the vectors generated by the USGS has reflected in the large processing times necessary for this task. It has taken their dedicated PDP 11/60 nearly five hours; this (they claim) is largely a result of the software being optimised for non-intersecting line (contour) processing (Smith pers comm 1981).

In general, the raster-to-vector conversion processes have been central processor-intensive, largely as a result of the high volume of data and complexity of the linework. This confirms the notions discussed in section 5.1 regarding slow-rates of processing using serial-operations (such as raster-to-vector) with raster data. The problems of software optimisation is important when classifying performance rates generally for the commercial systems. With the exception of the Kongsberg system, all those tested were claimed not to be tuned to handle the complexity of the OS interconnected lines. Clearly, it is impossible to assess the performance of these systems without adequate tuning towards OS data. Further tests - probably on a repayment basis - are necessary in which more extensive tuning and modifications to the software are carried out first. Thus in the light of the results gained from this research, it is concluded that there are no adequate commercial raster-driven systems suitable for the Ordnance Survey at the present time. The system operational at the USGS is adequate in terms of vector quality but provides an inappropriate data structure and its cost of raster-to-vector conversion may not be cost-effective. It is unreasonable to take c. 6-7 hours scanning and processing to generate a vector file which still requires data restructuring and feature-coding. Continual research is crucial to develop an effective system for Ordnance Survey needs.

7.5 DESCRIPTIONS OF RASTER-DRIVEN OUTPUT DEVICES

Computer-graphics display systems producing temporary-copy (from display screens) and hard-copy (from printer/plotters) are developing rapidly. The increase in technology and versatility of the temporary-copy devices is largely a result of the constantly decreasing price of semiconductor memory and the increasing sophistication of cathode-ray display tubes (Petrie 1981). In general, digital mapping systems are expected to require both temporary displays for interactive data manipulation and hard-copy plotters for the production of permanent records. There is some debate about whether vector or raster-driven output devices should be used for data presentation in existing commercial systems. Certainly a device capable of displaying raster information has the versatility to handle both vectors and rasters; the reverse does not necessarily produce a workable solution (see section 4.2.3). Systems from Kongsberg and Bell Laboratories have opted for immediate conversion of the scanned information to vectors for display on conventional plotters and Tektronix storage-tubes. Conversely, the approach by the Scitex Corporation has been to preserve data in raster notation and utilise refresh raster displays and laser film-writers.

It is evident that the nature of the data entry stage need not concern the methods of data processing and output

although the level of the results described in section 7.4 suggest that only limited transformations between rasters and vectors is possible at present. Thus, there is still a clear distinction between the uses of vector- and raster-driven hardware at present. Vector-based display strategies have been discussed widely elsewhere (Rhind 1977) and need no further explanation here. The use of raster-based techniques for data presentation will now be discussed since they have importance for the display and interactive manipulation of cartographic data in the medium-term future. Chapter five pointed out the increasing need for efficiency in handling these data due both to the anticipated data volumes and the need for fast and economical response rates; raster hardware has the potential of providing these facilities - especially when attached to dedicated micro- and mini-computers.

7.5.1 Raster-driven refresh graphics displays

The most important point in favour of these devices is their low price. The development of Cathode-Ray Tubes has received its main impetus from the television industry; indeed the lowest-cost raster graphics terminals contain standard television tubes. In general, a raster terminal has a heavy demand for semiconductor memory because each pixel on its screen has to be allocated a specific area of memory in

its buffer store. At present, the principal defect in most medium-priced devices for most cartographic work is their low resolution; typical display areas of 512 x 256 pixels are offered, which is well below the resolution offered by conventional vector-based storage tube displays. A Tektronix 4014 nineteen-inch storage tube, for example, offers up to 4096 x 4096 addressing, although resolution is only available to 1024 x 1024 - producing overlapping pixels. More sophisticated raster systems offer displays of up to 1000 lines although costs for these rise dramatically over their lower resolution counterparts.

The advent of colour has greatly increased the descriptive power of graphics and had most impact on the development of versatile, interactive systems in recent years (Petrie 1981). Sloan and Brown (1979) described general techniques for incorporating interactive, colour raster displays in a map production system. A highly adaptable interactive capability is offered on the Scitex Response 250 system using a nineteen-inch colour display screen of 320 x 256 points (Scitex Corp. 1981). This resolution (they claim) is adequate for the manipulation of graphics in all types of topographic and thematic mapping. All user interaction and data processing tasks are carried out on the controlling mini-computer within the Scitex system. An interesting development with considerable possibilities for digital

mapping is the multi-plane capability now available on some of the more expensive stand-alone raster displays. Most of the results displayed from the tests carried-out on the ICL DAP and reported in chapter six were produced on such a device - a Sigma R5680 display. Microprocessor control provides a graphic generator and four planes of associated store comprising 768 x 512 bits for use with the colour monitor (Sigma Electronics No-date). The multi-plane facility thus allows the storage of four wholly independent monochrome plots which can be either displayed singly or in any combination. In this way, for example, it is possible to assign different colours to specific classes of feature interactively. Areas (such as the Ordnance Survey sub-parcel data described in 6.4.2) can be displayed as multi-colours using polygon infill methods. Pavlidis (1979) has described a number of algorithms to perform area infill in raster-mode although the more sophisticated devices (such as the Sigma R5680) already have appropriate software inside their controlling microprocessor. It is unnecessary, therefore, to modify existing user-programs to utilise these powerful, interactive facilities. Current costs of these devices are in the range £5000 to £8000.

7.5.2 Raster-driven hard-copy plotters

The development of raster-driven hard-copy devices is becoming increasingly popular because of their speed and ability to produce graphics which are not restricted purely to line-work. Digital representations of photographic images and half-tone prints can be displayed quickly and easily. Current technology has adopted two distinct strategies: the optical exposure of light on to film and the electrostatic charging of special paper. In particular, the latter approach is popular for the rapid production of edit-plots. Versatec Electronics Ltd. manufacture (perhaps) the best known of these devices which are becoming commonplace in many areas of computer-graphics work. The method of operation is highly ingenious. A writing-head comprising many nibs (typically 100, 160 or 200 per inch) traverses a roll of electrographic paper and transmits an electrostatic charge for set-pixels in each scan-line. The paper then moves through a toning system in which special toner sticks to areas of the paper carrying the electrostatic charge. The paper is dried and ready for immediate handling. Versatec plotters can handle rolls of paper between 8.5 and 72 inches wide; each roll is available in rolls of 500 or 1500 feet in length. Since the electrostatic writing process is entirely electronic, the only movements in the plotters are in the paper transport and the toner supply. For these reasons,

plotting is fast and silent; typical plotting speeds can range as high as 1000 scan-lines per minute (c. two-inches per second). A solid-state buffering system ensures maximum speed plotting without loss of data. Such plotters are highly versatile for map-production, particularly for editing purposes; shading areas, half-toning and the production of grid patterns are all possible with relative ease.

Figures 7.22 and 7.23 show extracts of Ordnance Survey raster data plotted on a Versatec plotter with a resolution of 200 pixels per inch. Interfacing software was developed to enable raster files described earlier to be dumped directly on to the plotter. The displays were produced on an eleven-inch Versatec plotter installed in the Department of Geological Sciences at Durham University. The normal procedure used on the plotter at this site is to pass any vector files through Versatec interfacing software (Versaplot), which converts them to a raster form suitable for entry to the plotter; this software is addressed by easy-to-call subroutines which are attached to user's programs in a manner similar to the use of other plotting libraries such as Gino and *Ghost. A direct dump of a raster image, however, - as was carried out in this instance - was discovered to operate far faster than with the use of the Versaplot method (Stephenson pers comm 1981) since no vector-to-raster conversion was necessary first.



Figure 7.22: Extract of simulated
4096 x 4096 raster image of SO 9283SE
produced on Versatec raster plotter.
Crown Copyright Reserved.

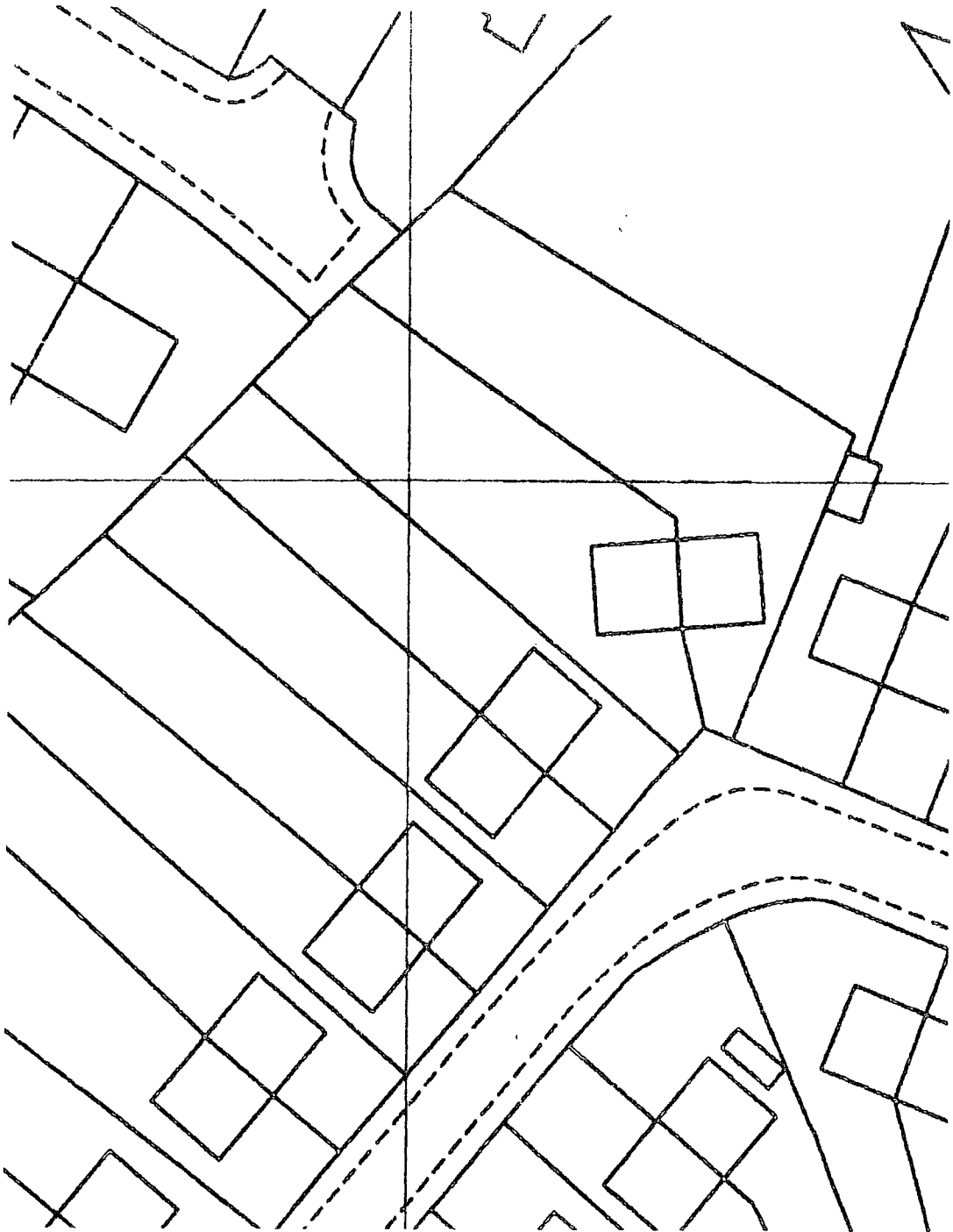


Figure 7.23: Extract of Scitex
8000 x 8000 raster image of SO 9283SE
produced on Versatec raster plotter.
Crown Copyright Reserved.

Figure 7.22 shows the results of a 4096 x 4096 pixel representation of 1:1250 sheet SO 9283SE after skeletonisation - each line has been reduced to one-pixel wide. Figure 7.23 shows an example of the 8000 x 8000 pixel raster data generated by the Scitex scanner at USGS prior to any skeletonising - each line is c. six-pixels wide. The results, although clear, are not to the standard attainable by the Xynetics plotters producing vectors (see section 2.5.3) which are currently used for editing purposes in-house at Ordnance Survey. Versatec plotting speeds, however, are impressive: the 4096 x 4096 data took c. 40 seconds to plot and the 8000 x 8000 data took c. 110 seconds to plot, compared with c. 20 minutes per OS 1:1250 map sheet on a Xynetics vector plotter.

Equally elaborate and highly accurate raster-driven plotters are available which operate on a principle similar to that of drum-type scanning devices. The plotters expose images on high-contrast, medium speed arts film (line or lith) which is secured to the rotating drum. Scitex Corp. use a laser-based method of exposure on their precision plotter which forms the data presentation component of the Response 250 raster-cartographic system (Scitex Corp. 1981). These plotters are highly suited to Landsat and similar data (e.g. such a plotter was used to plot wall-charts and grid-square census data in Britain (Rhind, Evans and

Visvalingam 1980). A continuously variable scanning resolution of between four and seventy-two pixels per millimetres is achievable on the Scitex device. Gronemann and Bar-Lev (1979) described the intricacies of this plotter and summarised its performance in a production environment. No appraisal-tests have been possible on this type of plotter although general discussions with various experts in the field (Scitex pers comm 1981, Smith pers comm 1980) suggest it to be appropriate for Ordnance Survey standards. However, more detailed appraisals would be desirable.

7.6 CONCLUSIONS

The ability to raster-scan existing OS map sheets is clearly within the bounds of current technology; a number of scanning devices are commercially available to perform this task quickly and easily. This judgement has confirmed the findings of a number of other research workers involved in similar feasibility studies (e.g. Robertson 1980, Boyle 1979, Davey et al 1977) who have drawn similar conclusions. However, the work reported here is the first attempt at using scanning methods to digitise Ordnance Survey mapping; in particular, the detailed large scale map data of the UK. These data comprise wholly line-work including pecks and text-characters. Most scanners - especially those designed

for cartographic data - can provide adequate pixel representations of these data. Their resolution capabilities and speed of operation are acceptable for mass-production purposes. Generally, scanning of a single 1:1250 plan has taken between thirty and seventy-five minutes; this elapsed time is dependent on the chosen resolution of the pixels (twenty pixels per millimetre is advised for the large scale OS-type mapping). Conservative estimates from these rates of scanning suggest that one device can convert between five and eight map-sheets to pixel form in each eight hour working day. It seems inevitable, however, that such expensive equipment may well operate on a two-shift per day basis, thereby generating between ten and sixteen sheets in raster form each day. Thus, assuming five scanners are available with c. 230 working days per year, then all 220,000 basic scale OS maps can be converted to pixel form in between ten and twelve years.

Of necessity, little investigation was possible to discover the extent of problems associated with poor quality source documents. It is to be expected that much image enhancement will be necessary to an unknown extent; complete redrawing will be necessary in exceptional cases. Clearly, further scanning-trials (which have been impossible in a research project of this sort) are required to qualify the problems. The use of photographic enhancement with special

purpose line film may offer possibilities here. Similar tests are being undertaken by the Ordnance Survey which is considering appropriate strategies to provide adequate source documents for its Laser-scan Fastrak digitiser (Latham-Wardepers comm 1981).

The extent of noise and coding-error in the raster file is tolerable for most software to detect and correct. Effective data validation is closely related to program tuning which suits the characteristics of the scanned data. This, in itself, is worthy of further research - raster images of dense contouring are markedly different to an urban 1:1250 plan, for example.

In general, then, the capabilities of existing scanning hardware are adequate for large-scales mapping. With the exception of the Scitex equipment, all scanners can handle monochrome images only. This does not hinder their suitability for these data since they are already held on monochrome sheets. However, the versatility of a scanner which can detect multi-coloured images in addition to black/white data should not be underestimated. It is possible to use such hardware for other roles in the map making process - as a front-end to the production of colour-separate information for printing plates or the single-pass scanning of multi-coloured small scale mapping

are examples. Further, complete raster scanning (in colour) of the 204 sheets in the Landranger (1:50,000) mapping has been quoted as being possible within c. 750 hours (36 pixels per millimetre resolution) by the Scitex Corp (King pers comm 1980). This would provide a national topographic data base in computer form, to serve as an interim measure for the UK until complete digital large scale coverage is available. Clearly, the extent to which this is commercial propaganda can only be assessed validly with some scanning trials for small scale maps.

The costs of these specialised devices compare favourably with similar automated line-following digitisers. Raster scanners which are suitable for large scale topographic cartography cost c. \$250,000; similar figures are quoted for a Fastrak device with associated controlling computer.

There is evidence to suggest that raster-driven output devices are equally adaptable in the mapping industry. Traditionally, the raster-based capabilities of this hardware have not been used to full advantage. Many of the manufacturers provide vector-to-raster software to facilitate the display of vector-based data but only those data-sets already structured in pixel form (e.g. Landsat imagery) have been able to utilise the raster-driven display capabilities to full advantage. This situation seems certain to change as

raster data in the mapping industry gain more impetus - when this time comes, raster-driven plotters will become more commonplace. In particular, the electrostatic printers have much potential for the production of edit-plots for graphics validation in a digital production flowline. Although the quality of their output is not to as high a standard as the sophisticated vector-based hardware currently used by the Ordnance Survey, their speed of operation is more impressive. Evidence has suggested that the raster-driven plotters can produce a 1:1250 map sheet ten times faster than the Xynetics vector plotters at the Ordnance Survey. In addition, the use of raster-driven display screens can offer the advantages of versatility and interactive graphics capabilities for map production. Their low-cost and extensive facilities (e.g. colour display and refresh image up-date) are expected to be highly functional in future production systems.

In general, it is concluded that the extent of hardware technology for all specialised raster input and output is sufficiently advanced for serious mapping applications. It is the status of the associated raster-based software and user-experience which is insufficiently developed at the present time. The problems of efficient and working algorithms to access and manipulate the digital raster-files automatically are prominent in all stages of the post hoc processing. In particular, the limitations of viable

raster-to-vector conversion have become evident in this work. The large scales data used in the tests are manifestly more difficult to convert to vectors than is the case with non-intersecting (contour-type) data. The large scales line work has a precise geometry which has been difficult to reconstruct successfully by software. Further, the structure of the resulting vectors is not appropriate for immediate entry into most data bases for other manipulation and archiving. Large amounts of data validation and cleaning are essential to generate a data organisation which is acceptable for the further stages of attribute tagging and interfacing with other data sets. Additionally, the amounts of central-processor time necessary for the raster-based procedures have proved to be lengthy for the complex large scales data. The only raster-to-vector system which has provided results of a display quality to the standard expected for Ordnance Survey work required five hours of processing on a dedicated PDP 11/60 computer.

All vectorising systems tested - with the exception of that operational at the USGS - have been unable to chain pecked-lines to produce solid line features. This has serious implications for the future of such an approach for OS operations. There is a difference between wishing to manipulate graphics as graphics and wishing to use graphics as topographic data. Clearly, software needs development to

string pecks together automatically (in a manner similar to the technique used by the Fastrak when digitising pecked-lines automatically). Pecks can easily be generated from solid lines but the reverse seems not to be trivial according to the scanning concerns talked-to. It seems intuitive that the DAP-type of architecture may well offer conceptually easy solutions to this peck-chaining, although no work of this nature has been attempted in this research.

On the whole, the results from existing commercial systems have been (somewhat) discouraging for serious production work. In essence, the time taken to generate a file by these systems, with a content and structure comparable to those emanating from a Fastrak digitiser is virtually the same (estimated at ten-twelve hours). A raster approach may still have serious implications in the data entry stages if the post hoc processing is made more efficient with a reduction in the costly and labour intensive roles of a human operator. The need for much applied effort is clear for raster systems to progress to a standard appropriate for OS requirements. Some systems are already sufficiently advanced to be serious propositions in a few years from now.

APPENDIX 7.1

Summary of the raster and vector data formats in-use by the
Bell Telephone Laboratories Inc.

(a) Run-length-encoded raster file:

Introduction:

The output from the EDAD scanner is in eight-bits pixel form. The number of pixels per line can be set by the operator according to the width of the drawing. Each scan-line of data is run-length-encoded and written on magnetic disk. The length of each scan-line (record) varies; this is dependent on the drawing density of the source graphic.

File structure:

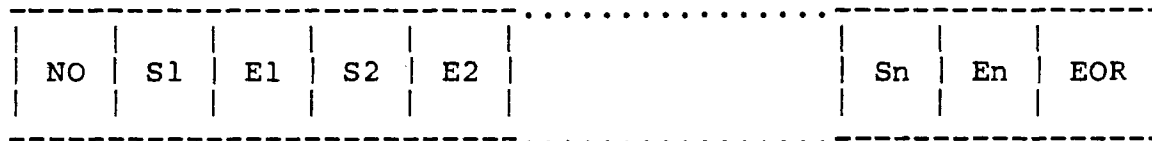
The file consists of 'n'+1 records. This number corresponds to 'n' scan-lines and one end-of-file (EOF) record.

Record structure:

The information stored in a record consists of header, data and trailer:

1. The header contains the length of the record.
2. The data contain the start and end addresses of the run-length pixel data.
3. The trailer is an end-of-record (EOR) marker.

Record Format



where: NO = number of fields to follow (n*2+1) (2 bytes)
Si = RUN i - starting address (2 bytes)
Ei = RUN i - ending address (2 bytes)
EOR = End of record; (-1) (2 bytes)

End-of-file (EOF) is a minus one (-1) written after the last data record.

(b) Vector data file:

File structure:

In the vector file there are 'n'+1 records. This number of records corresponds to 'n' vectors and one end-of-file record. All records have a fixed length of 14 bytes.

Record structure:

Each vector is described by seven field records:

TB	X1	Y1	X2	Y2	WD	TB
----	----	----	----	----	----	----

Where TB = total bytes in the data-portion of the record;
X1 = x coordinate of 'upper-left' corner of the vector
Y1 = y coordinate of 'upper-left' corner of the vector
X2 = x coordinate of 'lower-left' corner of the vector
Y2 = y coordinate of 'lower-left' corner of the vector
WD = width of the vector

Note: all fields in the record are two-bytes long

The last record in the file has X1 = -1.

8 CONCLUSIONS AND IMPLICATIONS FOR A NATIONAL DIGITAL RASTER
TOPOGRAPHIC DATA BASE

8.1 SUMMARY OF THE WORK

This thesis has described work carried out by the author to determine the feasibility of handling topographic data held on large scale maps by computer with a raster or grid-based format. Raster strategies for computer-assisted cartography offer serious alternatives to the more widely adopted vector-based techniques. In particular, raster data processing provides a second option for future Ordnance Survey planning, and this thesis has been especially concerned with raster methods for OS data and with the role which this alternative might play in future topographic mapping programmes in the UK.

The organisation of the thesis has been arranged in such a way as to introduce the concepts of a raster approach to mapping and to show the distinct differences to vector-based strategies. This introduction is then followed by a more detailed assessment of the likely characteristics and mechanisms necessary to manipulate effectively the large raster map files with conventional computing hardware. Since

it seems certain that new generations of computers will offer distinct advantages for the handling of large arrays of data, then the role that such systems might play in production has also been considered. In particular, the ICL Distributed Array Processor is suggested as a likely forerunner of future designs of computing systems. Extensive trials have been attempted using somewhat novel approaches to raster processing on the DAP and a section on this work has been included.

It is evident that most benefits from a raster approach to digital mapping will arise at the data entry stage; the conversion from paper- or film-based media to computer is far faster with a raster scanner than by any line-following mechanisms. For these reasons, a section of the thesis has been given over to descriptions and comparisons of commercial raster scanning systems known to be available in 1980/81. The ability to transfer data from existing images to a digital form with sufficient speed and quality of data with these systems is essential in any raster-based mapping system. In particular, scanning trials have been carried-out with OS data on all systems whose manufacturer's were willing to participate; a good response was received from most of the commercial concerns - all but one who were approached agreed to run the trials gratis.

8.2 IMPLICATIONS FROM RESULTS OF STUDIES INTO THE 'STATE OF THE
ART' OF DIGITISING SYSTEMS

The Ordnance Survey was probably the first major mapping organisation to introduce digital applications into the map-making process on a production basis; no other agency has so much experience of the problems and benefits of computer-assisted cartography on a large production scale. From this view-point, the work of the OS to-date has been exceptional: at present, c. 18,000 of the basic scale maps of Great Britain are held in computer form, generated by what must now be considered an out-dated digital mapping flowline.

Early assessments of the use of special-purpose interactive graphic editing systems for real-time data validation and correction by the OS are suggesting substantial improvements over the original batch editing system. Following the feasibility testing of one such system over a one-year period, the OS has now up-graded the facilities, essentially, to run five interactive editing stations (the Laser-scan IGES) in real-time from a Vax 11/780 computer. The use of these new mapping tools is considered successful for both interactive correction of existing digital files or for the creation of new files by a manual digitising process. It has proved straight-forward to train staff to operate these systems and the mechanisms are now

available to introduce the new facilities into the existing production flow-line.

Results from the Laser-scan Fastrak automatic line-following digitiser - the front-end to the OS development project TODDS (described in section 2.8) - have been rather less encouraging. Although it was envisaged that an automated line-following approach to OS map digitising would increase throughput rates dramatically (c. 5-fold), recent results have shown far less favourable speed increases with typical improvement factors ranging between two and four times, depending on the complexity of the data involved. This disappointing performance does not, of course, cast doubts on Fastrak's suitability for digitising other types of data. Intrinsically, the characteristics of the GB basic scale plans are not suited to the digitising role offered by a Fastrak; there are too many line-junctions connected by inappropriately short vectors for the device to operate without interruption for any length of time. The Fastrak is more suited towards expanses of uninterrupted line work, such as contour data or coastlines; further, recent work carried out by the author (not reported here) suggest its suitability for the high-speed digitising of some public-utility networks such as gas-mains, telecommunication lines, etc.

The evidence gathered in this research has suggested that all the raster scanning systems tested can generate efficiently a pixel-based digital image of OS large scale map data. These images can then be displayed directly on appropriate screens/plotters to provide further hard- or soft-copies of the maps or to provide a digital "back-cloth" on which to superimpose other data such as utility networks, census data, electoral and health statistics and so on. Clearly, though, the use of the data in OS mapping applications - and for suitable interfaces to existing vector-based GIS and manipulation/display systems such as the GIMMS and MOSS packages - dictate the need to generate equivalent data in vector notation. Further, it seems likely that adequate structure is required in the data to enable the inclusion and retrieval of specific feature codes and complete user-defined spatial entities, thus facilitating the flexible use of the resulting map-data archive for both graphic and non-graphic uses. It is the post hoc processes which necessarily must be applied to the scanned data which currently undermines the suitability of raster processing of map data for OS-type purposes.

The technical feasibility of introducing more structure into scanned data - whether it be by conversion to vectors, or the preservation of raster-cells - is still in need of much development. Glibly, most commercial agencies which

market raster devices describe the versatility and speed of their systems; all those systems tested in this project cannot be considered suitable for OS needs at present.

In particular, there is need for much further research and development into the effective handling of the following three map entities: line work - especially pecks, symbols and text. This research has necessarily restricted its investigations to the handling of map line-work; these data comprise the bulk of all information included in maps and are considered most difficult to process since even minor changes in a map's geometry of lines can alter completely the appearance and reliability of the working documents.

The handling of map symbols and text data in raster mode is not expected to be unduly difficult. Clearly, symbols on large scale maps are limited in extent although they do appear in the smaller scales; text is present in all mapping scales. The Scitex Corp. strategy for the handling of point symbols and text has been to produce their pixel formations in a raster pattern library for rapid recall at some later date. The positions in the map image of these text and symbols data are referenced uniquely and interactively by a trained operator on a special-purpose graphics editing station. Alternatively, Kongsberg Ltd. - which has opted for immediate conversion to vectors - suggests it to be most

cost-effective to convert the text and symbols to vectors along with all accompanying line-work and then to replace interactively the relevant vectors with an equivalent ASCII character or symbol macro on a special-purpose editing station. The introduction of specific type-faces can be predefined in a pattern library and recalled for subsequent plotting at a later date (c.f. Scitex Corp.). They (Kongsberg) suggest that automatic character recognition is in development within their organisation but not due to be released for some years yet. Clearly, however, some commercial systems are more advanced than others in this respect: recent modifications to those systems produced by Broomall Industries, for example, claim to perform effective character recognition (Logan pers comm 1982). Indeed, some completely new concepts in hardware seem likely to appear in the near future: hand-held cursors with local raster scanners and digitisers which can both follow lines or raster scan complete images are just two examples. It is evident that the results from a project of this nature cannot ever be completely up-to-date; the rapid developments in computer hardware designs and the ability of end-users to define more clearly their exact needs is enabling the creation of newer, more modern, more innovatory ideas and their introduction into working systems.

8.3 IMPLICATIONS FROM RESULTS OF STUDIES INTO POST HOC
MANIPULATION OF SCANNED DATA

A substantial proportion of the research content of this project has involved investigations of the mechanisms necessary and problems concerned with digital manipulation following the data entry (or scanning) stages. Clearly, there are already a number of "turn-key" commercial systems available for the handling of raster-type data.

It is clear, however, that none of the commercial raster systems tested offer complete solutions for mass throughput of the basic scales OS map series at the present time. All are costly in terms of computer processing time and storage and all require large amounts of interactive data validation and cleansing. No systems have provided digital vector data to a quality comparable with the existing Ordnance Survey flowline and no plotted results have been generated which meet current OS standards. Further, all line-work generated is held at one level with no included 'feature coding'. It has proved impossible to quantify the resources necessary to attach feature codes to this ad hoc line-work generated by the systems; this deficiency is largely a consequence of a lack of access to an appropriate data editing station for feasibility trials.

In essence, the nature of the results achieved have reflected the characteristics of the source data presented to the scanners. A series of graphics can only produce a digital representation of the same graphics. Thus, for example, most of the systems tested have preserved the pecked lines as pecks in the resulting vector files. In order to process these data for many other uses - as well as for purely graphic reproduction - it is necessary to string pecks together to generate the graphic line work in a more logical and useful form. In attempting to attach feature codes to the series of map lines and by reconstructing solid lines from the pecked data, we are, thus, striving towards a working information system and not merely an archive for graphic data.

The absence of any appropriate feature coding and, indeed, the poor structure of the line-work which is generated by most of the commercial systems is a major drawback to the use of a commercial raster system in production at present. Some systems can distinguish lines by either colours or by thickness but neither of these factors have much relevance for large scale maps which are monochrome (apart from the brown contour lines on the 1:10000 scale sheets) and have a limited variation of line-weights in their specification; there are clear indications that the smaller scale mapping series offer data which are more suitable for existing raster systems

although this has not been tested in practice.

The lack of structure and the classification of features being generated by the commercial solutions posed the need for research into other, more novel, strategies for post-scanning computer manipulation. To some extent, the results reported from the use of the ICL Distributed Array Processor - a new generation of computing hardware - has offered some first indications of the likely level of results possible in the future. It has been shown, for example, how a useful and logical structure can be introduced to raster-scanned data quickly and easily with a DAP method of processing; such useful structures and entity dissemination is seemingly difficult to achieve on the raster systems which were tested. The speeds of operation for certain tasks have been very encouraging on the DAP which operates most effectively on parallel operations. Thus, for example, the ability to sub-divide a complete binary raster image of a large scale map into a collection of sub-files - each comprising a single spatial entity - and other simple image manipulations such as scale changing and generalisation have been found to operate extremely quickly. Conversely, the ability to convert the boundary of these spatial entities into vectors to generate a file of polygons has been found to operate rather more slowly, since raster-to-vector conversion is, essentially, a sequential, line-following process.

However, speed of operation alone is not the only benefit to be concluded from a parallel approach to raster manipulation though, clearly, the large volumes of data inherent in raster structures do dictate the need for high speeds of operation. The DAP also has proved to be a versatile image processing tool in its own right. Many of the algorithms attempted in this research would be difficult to implement on a conventional serial processor. The ability to operate on complete sub-divisions of a total image makes for easier and conceptually more straightforward programming.

There are, of course, many constraints on the use of the DAP in work of this nature; it cannot be considered as providing a complete solution to raster-based digital mapping at present. The two dimensional connectivity of the processors in the architecture of the hardware means that algorithms perform most effectively on complete uncompressed raster images. Thus, the volumes of data being processed on the DAP in this work have been necessarily large - up to five times larger than the most effectively compacted raster files and up to 48 times larger than an equivalent vector method of storage. These figures are achieved even after the use of the efficient hierarchical data structure developed, which includes a substantial compaction over full raster arrays. Clearly, it seems intuitive to archive the data in some form more appropriate to a data bank and then expand them - or

even convert between vector and raster as required - for easier manipulation.

Secondly, the vector structure generated by the DAP after the conversion of the raster spatial entities into vector polygons is not ideal for subsequent processing, which is more likely to require segments extending between junctions or nodes. The vectors being generated in this work are presented as complete polygon boundaries with duplicated segments from polygons which are adjacent to each other. It is necessary - and possible - to process these vectors further to derive a link and node structure for more conventional processing by removing coincident lines and adding appropriate left and right feature coding.

Finally, since the DAP can presently operate only in batch mode, it has no capabilities for real-time interaction between the user and the machine; this is essential in any versatile image processing. Since the DAP used in these tests is part of a large mainframe computer which provides a complete university computing service, then the turn-around of batch jobs and, in particular, the speeds of graphics displays on raster terminals is poor. It is envisaged that this mode of operation for the DAP is only a first step; it seems likely, for example, that ICL will develop a stand-alone minicomputer configuration based on the ICL Perq

which will include a scaled-down version of the DAP in the near future (Smallbone pers comm 1982). Such an installation will provide a much improved tool for image processing tasks which require near instantaneous responses and fast, high-resolution graphic display capabilities.

8.4 IMPLICATIONS FOR THE NATIONAL TOPOGRAPHIC DIGITAL DATA BASE

It has been suggested that raster-based representations for map data are more suited to certain areal manipulations than are data in a vector form. Cellular picture processing, in general, requires shorter programs and more straightforward retrieval operations to extract specific regions within map images. Furthermore, it seems likely that a raster data structure for either a single map sheet or, indeed, when projected for the whole country will be less complex conceptually than will a segment and node vector structure. However, arrays of very small picture elements generate catastrophic storage problems for computer disk or tape media and are likely to overflow even the mass storage mechanisms currently being announced.

In particular, two strategies for storing raster data have been suggested in this work. The first is a run-length-encoded form of compaction in which superfluous

data are compressed on a scan-line by scan-line basis. The second storage strategy suggested has formed a hierarchical structure in which the data are compressed into two-dimensional sections: a low resolution indexing array provides pointers to higher resolution base data held in an uncompact raster form. On the basis of the work carried out in this research - in which only a small number of 1:1250 scale map sheets have been considered for test purposes - it is evident from the research that a scan-line run-length-encoded compaction scheme provides greater levels of data compaction than does the two-dimensional hierarchical scheme. The latter, however, provides better organisation and the ability to retrieve specific regions of data quickly and easily by virtue of its comprehensive indexing facilities.

Although caution must be exercised when interpreting extrapolations from such figures owing to the small sample of data currently held in raster form, it is possible to project the likely size of the complete topographic data base if held in raster form for the whole of Great Britain. Table 8.1 itemises these estimated projections on the basis of experience gained on the characteristics of a raster representation of OS basic scales data. It was shown in chapter five how even the smallest of variations in the resolution of each pixel will alter the size of the resultant

raster files dramatically, and hence these figures will change extensively as pixel resolution is varied.

Table 8.1 Estimation of the likely size of the GB basic scale topographic data base

No. of map sheets in series	217,777	
Vector storage in DMC character format	16×10^3 *	(200)
Uncompressed raster storage	3653×10^3	(38800)
Run-length-encoded raster storage	140×10^3	(1500)
Hierarchical raster storage	562×10^3	(6000)

*Source: Adams and Rhind (1981)

All storage quoted in megabytes for a 0.1 m pixel resolution. Figures in brackets represent estimated number of standard 6250 bpi, 2400 feet computer magnetic tapes.

Such large volumes of data as those envisaged in table 8.1 suggest it to be inappropriate to store the complete national data base in a raster form. It is manifestly more convenient to store the data as vectors for efficient use of computer backing store and for faster retrieval access times. Yet, in order to take advantage of the benefits of raster-based processing and both vector- and raster-based methods of input and output, it is essential to have the ability to interchange fluently between vector and raster as required. However, though it has been shown that vector-to-raster

conversion is readily possible, the reverse is not to a standard adequate for OS data in a production environment at the present time. Indeed, there are few systems available currently which perform routinely vector-to-raster and raster-to-vector conversions with fluency such that the task is ignored as being no longer a problem. In essence, a "black-box" interface is required in the future - whether it be a hardware, or a software or a hybrid solution - which can perform this conversion routinely leaving programmers free to concentrate on more applications-based software.

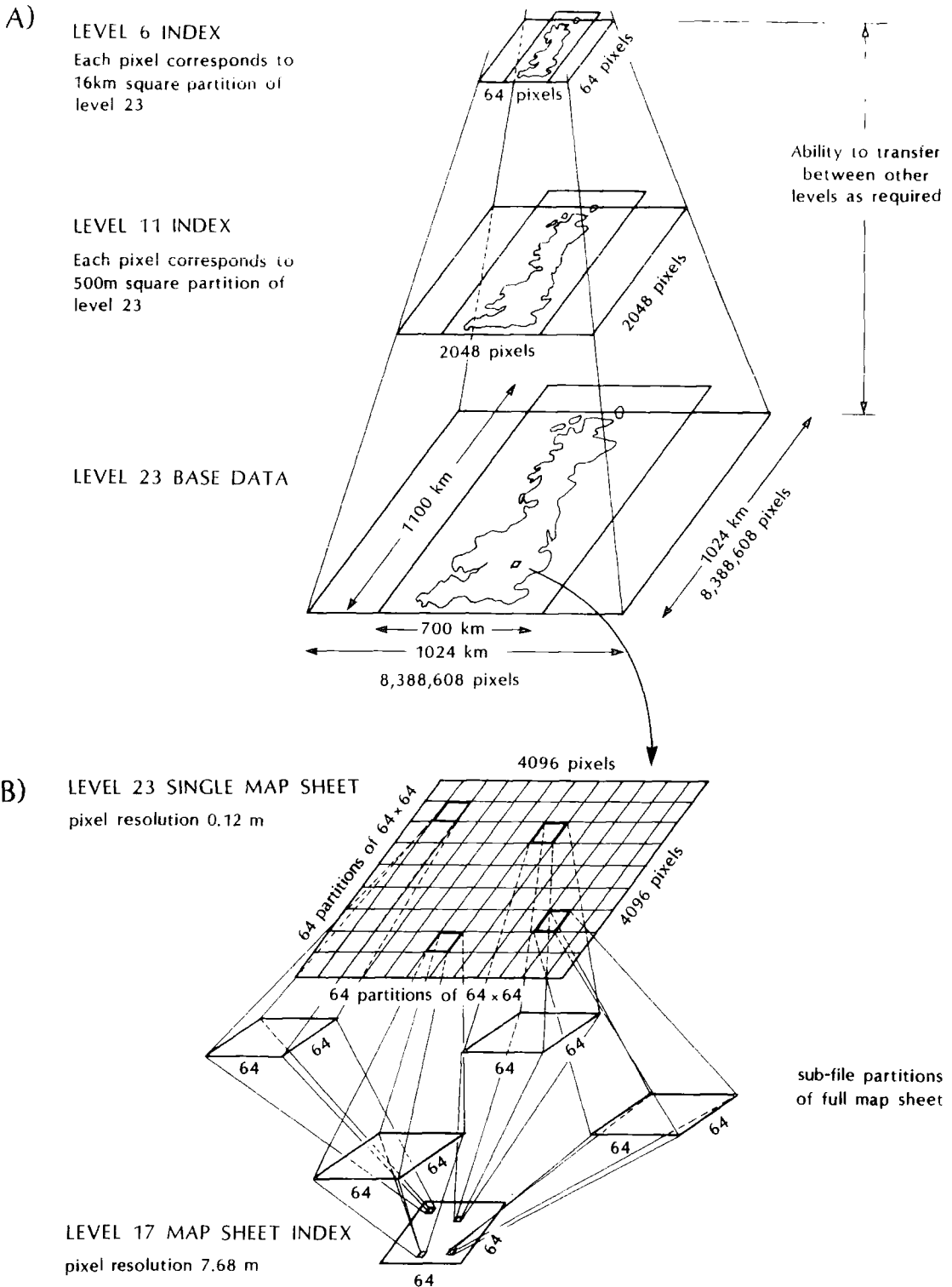
In conclusion, it seems likely that the volumes of data involved with a vector representation for the whole country are more manageable at present than with a raster mode. Nonetheless, it is not beyond the bounds of technical possibility to manage a raster data structure for the whole country. The extent of data involved is no larger than remote-sensed data handled routinely by the NASA EROS centre in Sioux Falls, for example, and a raster strategy to mapping has distinct advantages with the ease and speed of any input and display of the data.

8.4.1 A raster-based data base

Clearly, a raster data archive for the whole of Great Britain must at a minimum possess sufficient structure and organisation for rapid retrieval and up-date. It is suggested that the hierarchical data structure developed for DAP processing (described in detail in section 6.3 and figure 6.5) is eminently suitable to cover the whole country.

The notion of holding two levels only in this hierarchical data structure - an index and the base data - has been a preliminary experiment to investigate the potential of such an organisation for the data. It was pointed-out in section 6.3 that conceptually the hierarchy can be extended into any number of levels, depending on the range of pixel resolutions needed for a particular task - e.g. to display a complete map image on a 512 x 512 pixel raster display screen. Clearly, the volumes of data comprising a raster data base for the whole of Britain will need to take advantage of the multi-level capabilities inherent in this structure; it is certain that a multi-level of indexing to the base data will be required. In essence, there will be a series of hierarchical indexes: a low resolution index to the next higher resolution, which is itself a low resolution index to the next higher resolution level and so on. Such a data

structure will provide both adequate indexing to more detailed base data and a means of partitioning the base array of pixels for the whole country into more manageable units. Further, the intermediate levels of data (between the base level and the coarsest level index) will provide base data at varying resolutions for more generalised approximations of the total data as well as a series of intermediate index files. In theory, it will be possible for users to enter the data structure at any desired level and extract just the resolution of information which they require. Figure 8.2 shows some suggestions for the partitioning of a raster data base for the whole of Great Britain; in particular, figure 8.2a concentrates on the partitioning of the master files for the whole country into smaller units and figure 8.2b shows similar techniques for the sub-division of individual map sheets for manageable in-core processing. The National Grid referencing system for Britain comprises an area of 0 - 700 Km in eastings and 0 - 1100 Km in northings (excluding the Shetland Islands). Thus, to hold a raster image for this area with a pixel resolution of 0.1 mm at map scale (0.12 m ground scale) requires an array of pixels of dimensions $(5.7 \times 10^6) \times (9.0 \times 10^6)$. Clearly, it would be unrealistic to process such a large array without adequate partitioning; this is possible with the hierarchical concepts described in section 6.3. Holding the complete country with 0.1 mm resolution for the data hierarchy would entail rising



Suggested Data Structure For A UK Raster Topographic Data Base

Figure 8.2

to levels 23 or 24. Level 24 provides a raster of 16,777,216 pixels square - an areal extent of 2048 x 2048 Km on the ground. Level 23 provides a raster of 8,388,608 pixels square which represents 1024 x 1024 Km on the ground; the use of this latter level will offer dramatic storage reductions over level 24, although the Orkney Islands will be cut in half geographically at northing 1024 Km N. It is reasonable to suppose that alternative mechanisms can be used for the Orkneys and the Shetlands - perhaps as an insert in the digital files as is carried-out routinely by conventional cartographic means on small scale maps.

Thus, figure 8.2a shows the hierarchy necessary for the whole of GB and assumes that complete coverage of the country is held at level 23. The positioning of the country's NG reference frame within the level 23 base file is such to ensure that 1:1250 and 1:2500 scale sheet edges can - where possible - be overlaid with valid northings and eastings lines on the base array. It is evident, from figure 8.2a that there are redundant pixels in the eastings direction in areas of sea; these unused pixels could be wasteful of space although need not be stored in level 23 since references in index files held higher-up the hierarchy will show them to be empty pixels. This is one example of the benefits offered by the use of a series of hierarchical index files. Similar advantages will exist if the data are organised into a series

of binary files - one for each feature type - since those areas in which none of a particular feature exists need not be stored at level 23; for example, the 1971 population census of GB showed that only three-fifths of the country's area is populated (Rhind pers comm 1982) thus, it is reasonable to assume that there are many regions of the country in which no buildings are present.

The positioning of the indexes within the hierarchy should reflect the uses to which they will be put. For example, each pixel in a level 6 (64 x 64) index will represent 16 Km at ground scale (or a 131,072 pixel square in the level 23 base data); this, then, will provide only a crude descriptive summary of the data held in the lower levels. A level 11 index (2048 x 2048) will provide a more appropriate descriptive summary since each pixel in level 11 will represent 4096 x 4096 pixels in the base data or a 500 x 500 m square at ground scale which corresponds to a single 1:1250 scale map sheet.

For operational convenience, it is likely that the sheet edges of the large scale maps will remain important for some time yet. In particular, the scanning stages of the data base creation will be carried out on a sheet-by-sheet basis. Thus, it is essential to be able to position the resulting pixel files into their exact spatial locations within the

level 23 base data for the whole country. The level 11 index will provide a mechanism for achieving this and also show - near instantaneously - the progress of scanning of the map sheets as more and more are converted to pixel form.

It is likely that a data base (at level 23, say) will be handled (more appropriately) as a series of sub-files - one for each map sheet unit. Thus, a certain amount of compaction is possible by extending the hierarchical concepts to each raster map file. Figure 8.2b shows an example of the manipulation of a single map sheet in hierarchical terms relative to the complete raster structure for the whole of Great Britain. In essence, these concepts are an extension of the DAP-based data structure for single sheets developed in this research and described in section 6.3. The basic partition of each 4096 x 4096 complete raster file comprises 64 x 64 pixels; this is a dimension of convenience which is sufficiently small to be handled effectively on even the smallest of computers. An index to these partitions will then be a 64 x 64 pixel representation for one complete map sheet (there are 64 x 64 partitions of 64 x 64 pixels in a full 4096 x 4096 raster map sheets). The index, therefore, will be held at level 17 in the complete data base since each pixel will correspond to 64 x 64 pixels in the complete map image and will have a resolution of 7.8 ground metres.

This mechanism for a raster data structure has much potential for both the rapid storage and the flexible retrieval of individual map sheets - or indeed of small sections of individual map sheets - in the complete national data base; moreover, it provides a means of sub-dividing the complete data archive of pixels into partitions appropriate to the limited amounts of main memory offered by even the smallest of computers. Such a data structure - although ideal for the DAP - is not restricted to a parallel processor. In theory, it is possible to handle this form of data organisation on all sizes of serial processor too, though a smaller machine will require the conversion of the base data to very small partitions to enable any in-core processing. It follows that micro-computers with very limited amounts of main memory will be able to perform any operations on a similar basis to a mainframe, albeit more slowly, since they will have to repeat the manipulative operations a greater number of times on a larger collection of very small raster partitions.

8.4.2 The interrelation between vector and raster data in the data base

It is clear that users of digital topographic information require their data in both forms: vectors for linear

applications and pen-plotting and rasters for areal applications and for quick and easy methods of data entry and display - especially for background information on which to attach other variables. There are display terminals now available such as those emanating from Sigma, Megatec and GEC-Marconi, for example, which are capable of presenting both raster and vector data together and hence the possibility of overlaying OS topographic raster data held at one level (no feature codes) with network, satellite or other more thematic variables is immense.

Figure 8.3 is based on the work reported in this thesis and summarises the structure and components of a complete raster-based topographic data base system; it incorporates the DAP where necessary and attempts to show the need for an efficient conversion between vector and raster and the inclusion of suitable structures and feature categorising for all known user end-needs. In essence, attempts have been made in this research to develop prototype solutions to many of the stages suggested in figure 8.3. In particular, encouraging results have been gained in the data entry stages - scanning, data cleaning and reorganisation and sub-parcel identification. It is likely that further effort is required to develop a reliable method of automated/semi-automated spatial feature coding of the sub-parcel regions, although the initial results gained have shown promise. It is

THE BASIC COMPONENTS OF A RASTER - BASED TOPOGRAPHIC DATA BASE SYSTEM USING PARALLEL PROCESSING TECHNIQUES

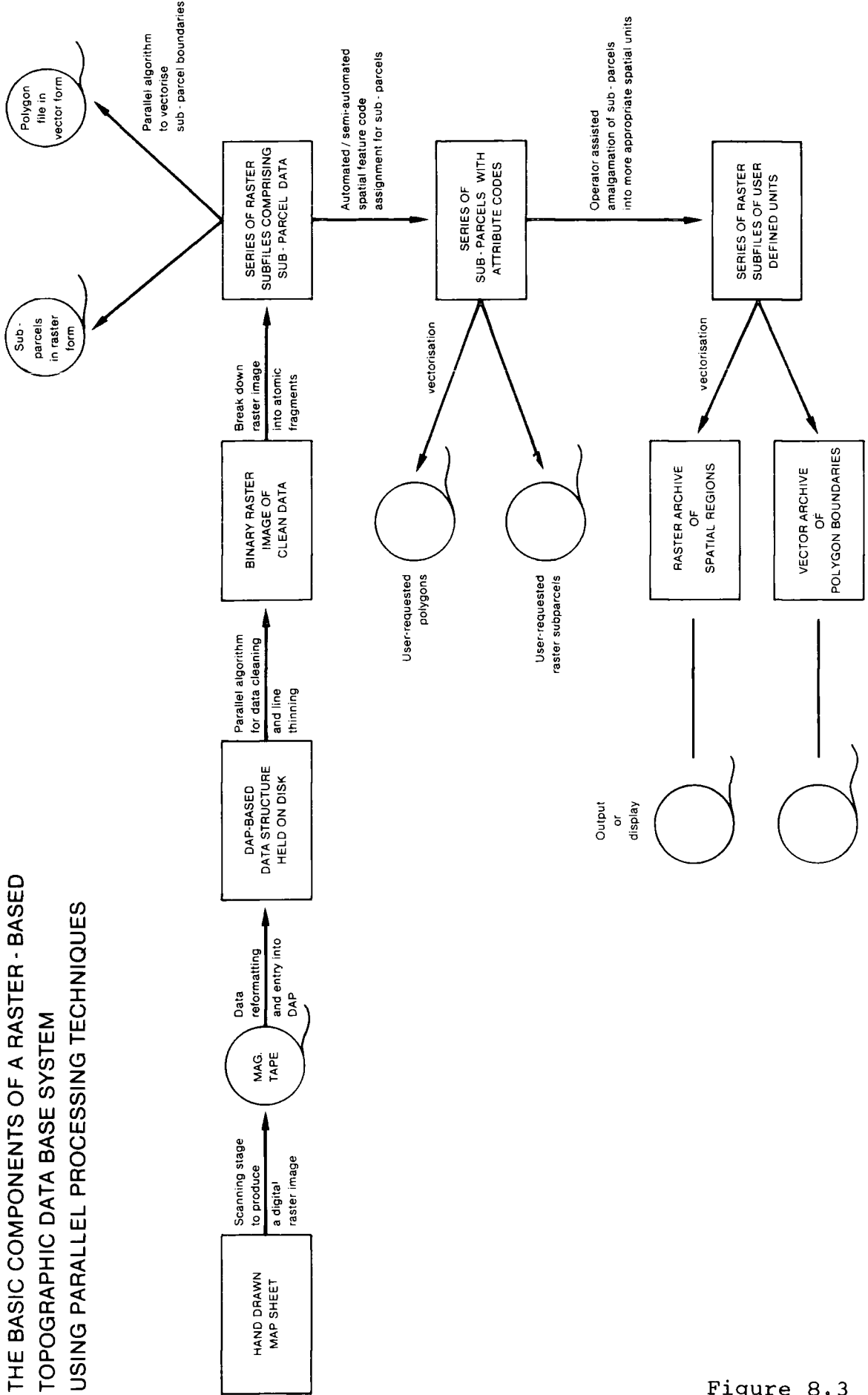


Figure 8.3

expected that a working prototype feature coding system could be tuned to suit OS data along the lines described in section 6.4.2. The ability to amalgamate the sub-parcel regions into more useful spatial units is expected to require large amounts of human assistance at present.

It seems likely that a mixed-mode data base combining the benefits of raster and vector notations is the most appropriate long term objective. With the exception of data capture at speed, most vector-based manipulation and storage is sufficiently well-advanced now to provide a mechanism adequate for routine storage and retrieval. Although a raster approach has advantages of neighbourhood searching and other areal based activities, it is costly to store for any sizeable geographic area. The volumes generated with vector-based equivalent files are substantially smaller but are more complex, with the need for lengthy cross-referencing between nodes and links files, text files and so on. It is inappropriate to store both vector and raster versions of the same geographic areas; this will be counter productive - especially in storage requirements and up-dates. It is more convenient to store one type of data and convert between vector and raster as required.

The quality of the source graphics is expected to have much control over the mode of entry of the data into digital

form. For operational reasons, it is likely that some poorer quality images are unsuitable for scanning and will, thus, be digitised manually by hand. It will be necessary, therefore, to develop appropriate interfaces between vector and raster methods of input to generate a single data base with all sheets held in a common notation. Clearly, it is possible to rasterise the vectors once they have been digitised and it is then possible to insert each digital map into a raster structure as though it had originated from a scanner in the first place. Since vectorising techniques are still insufficiently advanced, then the reverse approach (vectorising scanned data) is likely to be rather more troublesome at present.

8.5 SUGGESTED AREAS OF FURTHER STUDY

It is evident that a research project of the nature reported in this thesis is limited necessarily in the time and the expense which can be incurred during the work. For these reasons, there are several specific work areas which have been ignored although they are in need of further investigation for a comprehensive assessment of raster mapping. It is clear that a continual monitoring of the seemingly endless new systems and associated conversion software being developed by industries and by research

establishments for raster-based graphic data is essential. The hardware and software solutions being developed by these establishments are becoming more reliable, versatile and advanced quickly.

It is suggested that an appraisal of the new systems is only reliable to the degree required for investment decisions involving millions of pounds if specific scanning and post hoc processing trials are made and monitored carefully by both the professionals and the users. In particular, a high-priority should be given to tests which appraise the ease of use and the versatility of the complementary raster work stations. These have been mentioned several times in this thesis but were not investigated in detail because of the lack of access by the author to relevant systems. It is important, for example, to discover the problems involved in attaching the Ordnance Survey scheme of feature codes to a binary raster image (or a vector file with no attributes included): apart from the automated feature coding experiment (section 6.4.2), no consideration was made in this thesis of the best procedures to attach feature codes. Such a study can only operate effectively by including both experienced personnel who are familiar with the intricacies of OS feature coding conventions and with the operation of each particular computer-controlled work station. For these reasons, it was not considered practicable to attempt such a task in this

research although it is realised that an idea of the costings and mechanisms necessary for interactive editing are essential for a production raster system. During such tests, it will also be necessary to monitor the likely advantages and difficulties in handling text data on these systems. This too was not considered in great depth in this work due to similar reasons of lack of access to appropriate hardware.

All scanning trials and raster data used in this work have involved the use of a high-quality original document; this was considered a suitable first-step for handling OS data in raster mode. Clearly, the quality of all source documents in a production system are unlikely to reflect such a high-quality with high contrasting lines on a stable, robust, clean base. Moreover, the documents scanned in this study had text and stipple infill removed - this is likely to be impossible when scanning large numbers of sheets. A detailed investigation into the various conditions of the master survey documents being used in the OS regional offices is required, together with some considerations of the likely mechanisms for cartographic enhancement where necessary. In principle, it will be necessary to perform a series of scanning trials with a collection of MSD's of various qualities to discover the likelihood of handling all images directly on a raster scanner. It seems clear that some poorer-quality source documents will be of little use for

raster scanning and will probably require complete redrawing or manual digitising by vector means and subsequent rasterising.

Finally, whilst much experience has been gained on raster methods for large scale maps, it seems likely that existing technologies are more suited towards the smaller scales mapping series. It would be beneficial to attempt a similar preliminary study with OS small scale maps to assess the likely advantages and characteristics of a small scales raster data base.

BIBLIOGRAPHY

- ADAMS T.A. (1979)
Characteristics of a National Topographic Digital Data Base, Unpublished M.Sc. dissertation, Univ. of Durham.
- ADAMS T.A., MOUNSEY H.M. and RHIND D.W. (1980)
Topographic maps from computer output on microfilm, Cart. Jl., 17, 1, 33-39.
- ADAMS T.A. and RHIND D.W. (1981)
The projected characteristics of a National Topographic Digital Data Bank, Geogr. Jl., 147, 1, 38-53.
- ADLER R. (1978)
The organisation and structure of digital terrain models, Proc. symp. of FIG and Land Information Systems, Darmstadt, Oct. 1978.
- ALDRED B.K. (1972)
Points in polygon algorithm, IBM Scientific Centre Report UKSC-0025, Peterlee, UK.
- ALLEN R.R. and MORRIS L.P. (1980)
High performance interactive graphics, Proc. Harvard Graphics Week, Cambridge, Mass., July 1980.
- AMIDON E.L. and AKIN G.S. (1970)
Algorithmic selection of the best method for compressing map data strings, Comm ACM, 14, 769-774.
- ARMS A. (1970)
MAP/MODEL System: System description and user's guide. Bureau of Governmental Research and Service, U. of Oregon
- Lord AVEBURY (1978)
The impact of computers on society seen from the western viewpoint. Paper read at 8th Conf. of the Australian Computer Soc., Canberra.
- BABCOCK H.C. (1978)
Automated cartography data formats and graphics: the ETL experience, Amer. Cart., 5, 1, 21-29.
- BARRETT E.C. and CURTIS L.F. (1976)
Introduction to environmental remote sensing, Chapman and Hall.

- BAXTER R.S. (1976)
Computer and statistical techniques for planners,
Methuen.
- BAXTER R.S. (1980)
A computer file and data structure for manipulating
digital outlines of polygons, Geo-Processing, 1, 243-255.
- BENNY A.H. (1979)
Coastal definition using Landsat and potential
bathymetric applications, Proc. Remote Sensing Society
Annual Conf., Dundee.
- BICKMORE D.P. and BOYLE A.R. (1964)
An automated system of cartography. Tech. Symp. Int.
Cartogr. Ass. Edinburgh
- BICKMORE D.P. and MOLINEUX A. (1978)
Land-use mapping by Local Authorities in Britain,
Experimental Cartography Unit, NERC.
- BLACK W., CLEMENT T., DAVEY P.G., HARRIS J.F. and PRESTON G.
(1979) The use of an interactive high-precision scanner for
image processing, Inst. Physics conf., ser. no.4 ch. 8
- BOMFORD A.G. (1979)
The role of a national mapping organisation. Proc. Conf.
of Commonwealth Survey Officers. Cambridge.
- BOYLE A.R. (1979)
Scan digitisation of cartographic data, Paper read at
NATO ASI on Map Data Processing, Maratea, Italy.
- BRASSEL K., WASILENKO M. and MARBLE D.F. (1980)
Computer software for spatial data handling: cartography
and graphics. Volume 3 of series prepared by the IGU
for the USGS. Contract no. 14-08-0001-16560.
- BRAUNLEDER B. and KOBER R. (1981)
Parallel processing with an array of microcomputers,
Microprocessors and microsystems, 5, 6, 241-245.
- BROOMALL INDUSTRIES INC. (1980)
Holdall of technical specifications for Scan-Graphics
and associated Broomall systems. Mimeo. 700 Abbott Dr.
Broomall, Pa.
- BURNSIDE C.D. (1979)
Mapping from aerial photographs. Crosby, Lockwood
and Staples.

- CALKINS H. and MARBLE D.F. (1980)
Computer software for spatial data: full geographic information systems. Volume 1 of series prepared by the IGU for the USGS. Contract no. 14-08-0001-16560.
- CEDERBERG R.L.T. (1979)
Chain-link coding and segmentation for raster-scan devices, Computer Graphics and Image Processing, 10, 224-235.
- CHAMBERLIN D.D. (1976)
Relational data base management systems, ACM Computing Surveys, 8, 1, 43-66.
- CODD E.F. (1970)
A relational model for large shared data banks, ACM Computing Surveys, 13, 6, 377-397.
- COLES D. (1974)
Ordnance Survey digital mapping, in Dale P.F. (ed) Automated Cartography, BCS special pub. 1, London.
- COOK D. and MAXFIELD W. (1967)
The development of a geographic base file and its uses for mapping, Proc. URISA, Long Island.
- COPPOCK J.T. (1975)
Maps by line printer, in Davis J.C. and McCullagh M.J. (ed), Display and Analysis of Spatial Data, Wiley.
- COX N.J., ALDRED B.K. and RHIND D.W. (1981)
A relational data base system and a proposal for a geographical data type, Geo-Processing, 1, 217-229.
- CRAY RESEARCH INC. (1980)
The Cray-1 series of computers, Tech. Pub. 2240008 D, 1440 Northland Drive, Mendota Heights, Mn 55120.
- CROSFIELD ELECTRONICS (1981)
System specification for Magnascan 550/570. Crossfield Electronics, 776 Holloway Rd., London N19 3JG.
- DALE P.F. (1977)
The land surveyor and land information management, Proc. conf. of FIG, Sweden.
- DALE P.F. and HOLLWEY J.R. (1978)
Tasks for the surveyor in land information management, Proc. symp. of FIG and Land Info. Systems, Darmstadt.

- DAVEY P.G., HARRIS J.F., LOKEN J.G. and PRESTON G. (1976)
Operational experience with microfilmed rain charts.
Paper read at Int. Conf. on Computer assisted scanning,
April, Padua, Italy.
- DAVEY P.G., HARRIS J.F. and PRESTON G. (1977)
Automatic boundary digitising for the Department of the
Environment, Nuclear Phys. Lab., Oxford University.
- DAVEY P.G., HARRIS J.F., HAWES B., STOREY C. and WEST N.
(1979) A four-view automatic measuring system for
bubble chamber film. Inst. Phys. Conf., 44, 8.
- DAVIS M.J.K., PROCTOR D.W., ASHKENAZI V. and CROSS P.A.
(1972) The readjustment of the Retriangulation of Great
Britain and its relationship to the European
Terrestrial and Satellite networks, OS Prof. Paper NS24
- DAY A.C. (1972)
London FORTRAN course, The Athorne Press, London.
- DIELLO J., KIRK K. and CALLANDER J. (1969)
The development of an automated cartographic system,
Cart J1, 6, 1, 9-17.
- DoE (1976)
Parsed data set of British Local Authority Boundaries,
Building Research Establishment Note N138/76, October.
- DOUGLAS D. (No date)
Selected algorithms for the handling of vector-based
geographic data. Unpublished Mimeo, Univ. of Ottawa.
- DOUGLAS D. and PEUCKER T. (1973)
Algorithms for the reduction of the number of points
required to represent a digitised line or its caricature
Can. Cart., 10, 2, 112-122.
- DOYLE F.J. (1978)
The next decade of satellite remote sensing, Photogr.
Eng. and Remote Sensing, 44, 2, 155-164.
- DREWITT B. (1973)
The changing profile of the map user in Great Britain.
Cart. J1., 10, 1, 42-8.
- ENVIRONMENTAL SYSTEM RESEARCH INSTITUTE (ESRI) (1971)
Polygon Information Overlay System. CPO, San Diego, Ca.
- FERRANTI-CETEC GRAPHICS LTD. (1979)
Functional specification of the Cadastral and Land-Use
Mapping and Information System, Spec/FCG/79/6 Issue 1.

- FLOATING POINT SYSTEMS INC. (1979)
The array processor brochure, PO Box 23489, Portland Or
- FRANKLIN W.R. (1979)
Evaluation of algorithms to display vector plots on raster devices, Computer Graphics and Image Processing, 11, 337-397.
- FRASER TAYLOR D.R. (1980)
The computer in contemporary cartography. Wiley.
- FREEMAN H. (1961)
On the encoding of arbitrary geometric configurations, IRE Trans. Electronic Computers, EC-10, 260-268.
- GARDINER-HILL R.C. (1972)
The development of digital maps, OS Prof. Paper NS23.
- GOODYEAR AEROSPACE CORP. (1974)
System description of STARAN, Akron, Ohio.
- GRONEMANN U. and BAR-LEV A. (1979)
Laser printing. Proc. Soc. Photo-optical Instrumentation Engineers. Los Angeles.
- HAGGETT P., CLIFF A.D. and FREY A. (1977)
Locational methods, 2nd ed. Arnold.
- HARDY P.G. (1978)
The Laser-scan Fastrak automatic digitising system, Paper read at seminar on "Data banks and Digital Mapping", RICS/N.E. London Poly., Nov. 1978.
- HARLEY J.B. (1975)
Ordnance Survey maps - a descriptive manual, HMSO.
- HARRIS J.F., KITTLER J., LLEWELLYN B. and PRESTON G. (1981)
A modular system for interpreting binary pixel representations of line-structured data, Paper read at NATO ASI on Pattern Recognition: theory and applications Oxford, March 1981.
- HARRIS L.J. (1980)
The application of computer technology to topographical cartography, in Taylor D.R.F. (ed) The computer in contemporary cartography, 59-92, Wiley.
- HARRIS R. (1979)
Access to Landsat data, Area, 11, 1, 63-66.

- HARRIS R. and BARRETT E.C. (1978)
Towards an objective nephanalysis, Jl. Appl. Met., 17,
1258-1266.
- HARRISON J.C. (1978)
The LAMIS system, Paper read at seminar on "Data banks
and Digital Mapping", RICS/N.E. London Poly. Nov 1978.
- HARVARD UNIVERSITY LABORATORY FOR COMPUTER GRAPHICS AND
SPATIAL ANALYSIS (1974)
POLYVRT Manual. Cambridge, Mass.
- H.M.S.O. (1979)
Report of the Ordnance Survey Review Committee, London.
- HOWARD S. (1968)
A cartographic data bank for Ordnance Survey maps,
Cart. Jl., 5, 48-52.
- HOWMAN C. and WOODSFORD P.A. (1978)
The Laser-scan Fastrak automatic digitising system,
Paper read at ICA 9th Conf. on cartography,
Washington D.C., July 1978.
- ICHIKAWA T. (1981)
A pyramidal representation of images and its feature
extraction facility, IEEE Trans. on pattern analysis
and machine intelligence, 1-3, 3, 257-264.
- I.C.L. (1979)
DAP: APAL language, ICL Tech. Pub. 6919, 1st ed.
- I.C.L. (1981)
DAP: FORTRAN language, ICL Tech Pub. 6918, 3rd ed.
- KEATES J.S. (1973)
Cartography design and production, Longmans.
- KEIR K.M. (1976)
A digital mapping system designed for the commercial
air survey market, Paper read at 13th congress of the
ISP, Helsinki.
- KLETTE R. (1979)
Parallel operations on binary images, Computer Graphics
and Image Processing, 14, 145-158.
- KONGSBERG LTD. (1980)
Holdall of technical specifications of SysScan and
associated Kongsberg systems. Mimeo. Priors Way,
Maidenhead, SL6 2HP.

- LASER-SCAN LABORATORIES LTD. (1981a)
Laser-aid: the Laser-scan Fastrak user's guide, rel 7.
11 June 1981.
- LASER-SCAN LABORATORIES LTD. (1981b)
IGES: cartographic editing system user's guide, rel 2.
18 June 1981.
- LLOYD C. (1981)
The Kongsberg SysScan system for automated cartography.
Proc. Ann. Symp. Brit Cartogr. Soc., Southampton.
- LOGAN I.T. (1981)
Ordnance Survey digital mapping, Paper given at Conf.
on Survey and Mapping, RICS, Reading, April 1981.
- LOODTS J. (1981)
Vectorisation ou intelligence artificielle en
cartographie automatique, Bull. trimestrial de la Soc.
de Photogr., 141/142, March/July 1981.
- LOOMIS R.C. (1965)
Boundary networks, Comm. ACM, 8, 44-48.
- MacDOUGALL E.B. (1976)
Computer programming for spatial problems. Arnold.
- MALLER V.A.J. (1979)
The content addressable file store - CAFS, ICL Tech.
Jl., 265-279, November.
- MARBLE D.F. and PEUQUET D.J. (1978)
Problems in the storage and manipulation of large
spatial data sets, Proc. UNESCO Conf. on computer
mapping for resource analysis.
- MARBLE D.F. and PEUQUET D.J. (1980)
The Domestic Information Display System: some potential
developments, Mimeo.
- MARKS P. (1980)
Low-level vision using an array processor, Computer
Graphics and Image Processing, 14, 281-292.
- MARLES A.C. (1971)
The place of map miniaturisation in map production,
Proc. Conf. Commw. survey offrs., Cambridge.
- MARTIN J. (1976)
Principles of data base management, Prentice-Hall, NJ.

- McEWEN R.B. (1980)
The USGS digital cartographic applications program, Jl. of Surveying and Mapping Div. ASCE, May 1980.
- McEWEN R.B. and WHITE J.C. (1979)
Development of digital techniques for mapping and charting, Proc. Conf. Commw. survey offr., Cambridge.
- MINNESOTA STATE DEPARTMENT OF ADMINISTRATION (MSDOA) (1971)
The Minnesota Land Information System, Information Systems, Minnesota Natural Resources, Newsletter 2 March 1971, St. Paul, Minnesota.
- MITCHELL W.B., GUPTILL S.C., ANDERSON E., FEGAS R.G. and HALLAM C.A. (1977) GIRAS: A Geographic Information Retrieval and Analysis System for handling land-use and land-cover data, USGS Prof. Paper 1059.
- MUELLER C.G. (1970)
Some psychophysical and neurophysiological data on contrast and contour sensitivity, in Lipkin B.S. and Rosenfeld A. (ed) Picture processing and psychopictorics, Academic Press, London.
- NELDER J.A. (1975)
Computers in biology, Wykeham Publications, London.
- OPHEIM H. (1980)
A new method for data reduction of a digitised curve, Mimeo, Norwegian Computing Centre, Oslo.
- ORDNANCE SURVEY (1981a)
Annual Report 1980-81, Southampton.
- ORDNANCE SURVEY (1981b)
Summary report of the study of revision, Aug. 1981, Southampton.
- PAGE E.S. and WILSON L.B. (1978)
Information and manipulation in a computer, 2nd ed., Cambridge Univ. Press.
- PALMER A. (1980)
Computer aided land surveys. Chartered Land Surveyor/Mineral Surveyor Quarterly, Winter 1980, RICS.
- PARKINSON D. (1976)
Computers by the thousand, New Scientist, 17 June 1976.
- PARKINSON D. (1977)
An introduction to array processors. Systems International. November.

- PARKINSON D. (1978)
High-speed computing, Phys. Bull., 29, 464-465.
- PAVLIDIS T. (1979)
Filling algorithms for raster graphics. Comp. Graphics and Image Processing., 10, 126-141.
- PETRIE G. (1981)
Hardware aspects of digital mapping, Photogr. Eng. and Remote Sensing, 47, 3, 307-320.
- PEUCKER T.K. and CHRISMAN N. (1975)
Cartographic data structures, Amer. Cart., 2, 2, 55-69.
- PEUQUET D.J. (1979a)
Raster processing: an alternative approach to automated cartographic data handling, Amer. Cart., 6, 2, 129-139.
- PEUQUET D.J. (1979b)
A raster mode algorithm for interactive modification of line drawing data. Comp. Graphics and Im Proc. 10, 142-58
- PEUQUET D.J. (1981)
An examination of techniques for reformatting digital cartographic data, Cartographica, 18, 1 and 2.
- PEUQUET D.J. and MARBLE D.F. (1980)
Computer software for spatial data handling: data manipulation programs. Volume 2 of series prepared by IGU for the USGS. Contract no. 14-08-0001-16560.
- PFALTZ J.L. and ROSENFELD A. (1967)
Computer representation of planar regions by their skeletons, Comm ACM, 10, 2, 119-125.
- PFERD W. and RAMACHANDRAN K. (1978)
Computer aided automatic digitising of engineering drawings, Proc. IEEE. Conf. on Computer Software Applications, pp630-635, November 1978.
- PFERD W. and STOCKER G.C. (1981)
Optical fibres for scanning digitisers, The Bell Lab. Tech. Jl., 60, 4.
- PFERD W. and WHITE J.P. (1978)
Vector coding methods for compaction of scan-digitised line drawing images, Mimeo.
- PMA CONSULTANTS LTD. (1978)
System specification of the conversion of Ordnance Survey digital map data to a restructured form, February 1978, Horley, Surrey.

- POOCH U.W. and NEIDER A. (1973)
A summary of indexing techniques for sparse matrices,
ACM Computing Surveys, 5, 2.
- REEVES A.P. (1980)
On efficient global information extraction methods for
parallel processors, Computer Graphics and Image
Processing, 14, 159-169.
- RHIND D.W. (1974)
An introduction to the digitising and editing of mapped
data, in Dale P.F. (ed) Automated Cartography, BCS
Special publ. 1, London.
- RHIND D.W. (1976)
Towards universal, intelligent and usable automated
cartographic systems, ITC J1., 3, 515-545.
- RHIND D.W. (1977)
Computer-aided cartography. Inst. Brit. Geogr. Trans.
New series, 2, 1.
- RHIND D.W. (1979)
"State of the art" digitising of graphics: Fastrak and
Response 250, Paper read at Australian National
Univ. workshop on Geographical Information Systems,
December 1979.
- RHIND D.W. (1981)
Computer processing and mapping requirements of the
'ecological mapping' community project. Final report
prepared for the Environment and Consumer Protection
Service of the Commission of the European Communities.
Dept. of Geography, Univ. of Durham, U.K.
- RHIND D.W. and ADAMS T.A. (1980)
Recent developments in survey and mapping, in Brown E.H.
(ed) Geography - yesterday and tomorrow, Oxford U. P.
- RHIND D.W. and ADAMS T.A. (1981)
Coordinate data bases: availability and characteristics
Harvard Library of Computer Graphics, 13.
- RHIND D.W. and EVANS I.S. (1980)
The display of population census data, Harvard Library
of Computer Graphics, 16.
- RHIND D.W., EVANS I.S. and DEWDNEY J.C. (1977)
The derivation of new variables from population census
data, CRU working paper 9, Univ. of Durham.

- RHIND D.W., EVANS I.S. and VISVALINGAM M. (1980)
Making a National Atlas of Population by Computer.
Cartogr. Jl. 17, 1, pp 3-11.
- RHIND D.W. and HUDSON R. (1980)
Land-use, Methuen.
- ROBERTSON P.K. (1980)
A raster-scan cartographic digitising system,
Unpublished M.Sc. dissertation, U. of Saskatchewan.
- ROBINSON A.H. and PETCHENNIK B.B. (1976)
The nature of maps. Chicago Univ. Press.
- ROGERS A. and DAWSON J.A. (1979)
"Which digitiser ?", Area, 11, 69-73.
- ROSE D.J. and WILLOUGHBY R.A. (ed) (1972)
Sparse matrices and their applications, Plenum press.
- ROSENFELD A. (1969)
Picture processing by computer, Academic Press, NY.
- ROSENFELD A. and KAK A.C. (1976)
Digital picture processing, Academic Press, NY.
- SAMAT H. (1980)
Region representation: quadtrees from binary images,
Computer Graphics and Image Processing, 13, 88-93.
- SCHMIDT A.H. and ZAFFT W.A. (1975)
Programs of the Harvard University Laboratory for
Computer Graphics and Spatial Analysis, in Davis C. and
McCullagh M.J. (ed) Display and Analysis of Spatial
Data, Wiley.
- SCITEX CORP. (1981)
Response-250: computer graphics for cartography.
Mimeo. 304 Ave. Louise, 1050 Brussels.
- S.D.D. (1978)
Rural land-use information system project, Scottish
Development Dept. of the Fife region.
- SEARLE P.R. and WATERS R.S. (1979)
The Ordnance Survey 1:10000 and 1:25000 digital maps,
Proc. Conf. Commw. survey offrs., Cambridge.
- SEYMOUR W.A. (ed) (1980)
A history of the Ordnance Survey, Dawson.

- SHAPIRO L.G. and HARALICK R.M. (1980)
A spatial data structure, Geo-Processing, 1, 313-317.
- SIGMA ELECTRONIC SYSTEMS LTD. (No date)
Technical description and user manual for R5680 colour display. Church Street, Horsham, Sussex. RH12 3QN.
- SIMMONDS M.C. (1978)
The presentation of survey information on microfilm, Mimeo paper at Inst. of Reprographic Technology.
- SLOAN K.R. and BROWN C.M. (1979)
Colour map techniques. Comp. Graphics and Image Processing, 10, 297-317.
- STEPHENS M.J. (1980)
The USGS 1:2 million-scale digital data base, Mimeo.
- SUMMAGRAPHICS CORP. (1979)
User manual ID Data tablet/digitiser. Fairfield, Conn.
- SWITZER W.A. (1975)
The Canadian Geographic Information System, in Wilford Brickwood J.M., Bertrand R. and van Zuylen L. (ed) Automation in cartography, 58-81, ICA, Netherlands.
- TAFOYA B.R. (1980)
Raster revisited or raster-to-vector and vector-to-raster systems. SPIE, Vol 222, Laser scanning and recording for advanced image and data handling. pp 148-55.
- TAYLOR P.J. (1977)
Quantitative methods in geography, Houghton-Mifflin.
- TEKTRONIX CORP. (1974)
4014 computer display terminal user manual, Beaverton, Or., July 1974.
- THOMPSON C.N. (1978)
Digital mapping in the Ordnance Survey: 1968 - 1978, Paper read at ISP Commission IV, Ottawa, October.
- THOMPSON C.N. and SHREEVE S.T. (1978)
Ordnance Survey and Remote Sensing, Proc. Remote Sensing Society Annual Conf., Durham.
- THURBER K.J. and WALD L.D. (1975)
Associative and parallel processors, ACM Computing Surveys 7, 4, 215-255.

- TIENSTRA M. and KRAAN J.G. van der (1975)
Some figures about automated large scale mapping, in
Wilford Brickwood J.M., Bertrand R. and van Zuylen L.
(ed) Automation in cartography, 166-173, ICA, Neth.
- TOMLINSON R. (ed) (1972)
Geographical data handling, UNESCO 2nd symposium on
Geographical Information Systems, Ottawa, August 1972.
- TOMLINSON R., CALKINS H. and MARBLE D.F. (1976)
Computer handling of geographic data, UNESCO Press,
Paris.
- UNITED NATIONS (1976)
Status of world mapping (World Cartography 14). Dept. of
Economic and Social Affairs, UN, New York, 102pp.
- VISVALINGAM M. (1975)
Storage of the 1971 UK census data: some technical
considerations, CRU working paper 4, Univ. of Durham.
- VISVALINGAM M. (1976)
Indexing with coded deltas - a compaction technique,
Software Practice and Experience, 6, 397-403.
- VISVALINGAM M. (1977)
A locational index for the 1971 kilometre-square
population census data for Great Britain, CRU working
paper 12, Univ. of Durham.
- VISVALINGAM M., NORMAN M.J. and SHEEHAN R. (1976)
Data interchange on industry computer compatible tapes,
CRU working paper 6, Univ. of Durham.
- WALKER A.S. and WHITING B.M. (1980)
Microcomputers in surveying - applications and problems.
Paper read at Surveying Teachers Meeting, St. Annes Coll.
Oxford, December 1980.
- WAUGH T.C. (1980)
The development of the GIMMS computer mapping system,
in Taylor D.R.F. (ed) The computer in contemporary
cartography, 219-234, Wiley.
- WEBER W. (1980)
Automation mit rasterdaten in der topographischen
kartographie, Kartographische Nachrichten, Oct.

WOHLMUT P.G. (1976)

The i/o Metrics automatic digitising system. Proc. of the ACSM Int. Conf. on Automation in Cartography "Auto-Carto 1", Reston, Va. Dec 1974.

YOELLI P. (1977)

Computer executed interpolation of contours into arrays of randomly distributed height points, Cart. JI., 103-108.

LIST OF PERSONAL COMMUNICATIONS

BBC (1980)

Visit to Mr W. Gardner, Head of BBC Computer Graphics Unit, BBC TV Centre, London W12 on 14th Nov. 1980.

BLYTHE J.H. (1981)

Visit to GEC-Marconi Electronics Ltd. Communications Research Laboratory, Chelmsford on 20th Mar. 1981.

BOYLE A.R. (1981)

Discussions with Dr Boyle (Univ. of Saskatchewan) during the British Cartographic Soc. Annual Technical Symposium Southampton on 12th-13th Sept. 1981.

Broomall Industries Inc. (1981)

Visit to Mr K. Planty and Mr M. Lemmon at Broomall Industries headquarters, Broomall, Pa. USA on 28th July 1980

FORTESQUE F. (1981)

Discussions with Mr Fortesque, Asst. Director Reprographics, Ordnance Survey during visit to OS headquarters, Southampton in Sept. 1981.

HAYWOOD P.E. (1980)

Discussions with Mr Haywood, Senior Survey Officer, Development Branch, Ordnance Survey during visit to OS headquarters, Southampton in Oct. 1980.

HOWMAN C. (1982)

Telephone conversation with Mr Howman, Cartography Manager, MCE RE, Feltham in Mar. 1982.

HUNT D.J. (1980)

Discussions with Dr Hunt, ICL Research and Advanced Development Centre, Stevenage on 4th Nov. 1980.

KING S. (1980)

Visit to Mr King, Sales Director, Scitex (North America) headquarters, Bedford, Mass, USA on 31st Jul. 1980.

Kongsberg Ltd. (1981)

Discussions with Messers. Anderson, Kaasa, Lloyd and Miller during Kongsberg seminar on "Advanced Technology in Cartography" held at the Crest Hotel, Maidenhead on 7th July 1981.

Kummerley and Frey AG (1981)

Discussions with M. Brunner and M. Ficker during Scitex seminar on "Cartography with raster-based computer graphics" held at the Hilton Hotel, Brussels on 28th-29th Apr. 1981.

Personal communications

LATHAM-WARDE D.L. (1981)

Discussions with Mr Latham-Warde, Senior Survey Officer, Development Branch, Ordnance Survey during visit to OS headquarters, Southampton in Sept. 1981.

LINKLATER C. (1979)

Discussions with Mr Linklater, Supt. ClD, Cartography Divison, Ordnance Survey during visit to OS headquarters Southampton in Nov. 1979.

LLOYD C. (1980)

Telephone and letter communications with Mr Lloyd, Cartography Technical Support, Kongsberg Ltd., Maidenhead.

LOGAN I.T. (1982)

Discussions with Mr Logan, Principal Survey Officer, Development Branch, Ordnance Survey during visit to Birkbeck College on 29th Apr. 1982.

LOODTS J. (1981)

Discussions with Dr Loodts, IGN Brussels, during Scitex seminar on "Cartography with raster-based computer graphics" held at the Hilton Hotel, Brussels on 28th-29th Apr. 1981.

Ordnance Survey

Discussions with various personnel during several visits to the OS headquarters, Southampton in the period Sep. 1979 - Jun. 1982. Chief amongst them are: Messers Arnold, Arnott, Bell, Boyle, Fraser, Fortesque, Haywood, Latham-Warde, Linklater, Logan, N.S. Smith, W.P. Smith, Thompson, Wesley.

PARKINSON D. (1979)

Visit to Prof. Parkinson, Head of DAP Support Unit, Queen Mary College, London on 22nd Nov. 1979.

PFERD W. (1980)

Visit to Dr Pferd, Head of Office Planning Dept., Bell Telephone Laboratories headquarters, Whippany, NJ, USA on 29th Jul. 1980.

PRESTON G. (1981)

Visit to Image Analysis Group, Nuclear Physics Laboratory at Oxford University on 22nd Apr. 1981.

RAGG J.M. (1981)

Letter of 31st Jan. 1981 and subsequent discussions with Mr Ragg, Soil Survey of England and Wales, Derby during the British Cartographic Soc. Annual Technical Symposium, Southampton on 12th-13th Sep. 1981.

Personal communications

RHIND D.W. (1982)

Discussions with Prof. Rhind, Head of Geography Dept., Birkbeck College (Ph.D. supervisor) in May 1982.

Scitex Corp. (1981)

Discussions with various personnel of the Scitex Corp. in North America and Europe. In particular, with Messers. Fono, Herzka and King during Scitex seminar on "Cartography with raster-based computer graphics" held at the Hilton Hotel, Brussels on 28th-29th Apr. 1981.

SMALLBONE J. (1982)

Visit to Mr Smallbone, DAP Marketing, ICL London headquarters, ICL House, London on 3rd Mar. 1982.

SMITH D.G. (1980)

Visit to Mr Smith, Computing Manager, USGS National Centre, Reston, Va, USA on 23rd-25th Jul. 1980.

SMITH N.S. (1981)

Discussions with Mr Smith, Survey Officer, Development Branch, Ordnance Survey during visit to OS headquarters, Southampton in Sept. 1981.

STEPHENSON D.L. (1981)

Discussions with Mr Stephenson, Research Officer, Dept. of Geological Sciences, Univ. of Durham in Oct. 1981.

US Bureau of the Census (1980)

Visit to Mr F. Broome and Mr M. White, Geography Division, US Census Bureau headquarters, Washington DC on 22nd Jul. 1980.

USGS (1980)

Discussions with various personnel from USGS, including: Drs Anderson, Guptill and Messers Schmidt, Smith, Southard and Stephens during visit to USGS National Centre, Reston, Va, USA on 23rd-25th Jul. 1981.

WESLEY P.W. (1981)

Discussions with Mr Wesley, Manager of Development Branch, Ordnance Survey during visit to OS headquarters, Southampton in Sept. 1981.

WOODSFORD P.A. (1981)

Discussions with Mr Woodsford, Director of Laser-scan Laboratories, Cambridge during British Cartographic Soc. Annual Technical Symposium, Southampton on 12th-13th Sep. 1981.

ANNEX 5a

FORTRAN source listing of program FROMDMC see section 5.3.3

```
1 C***** T H I S I S F R O M D M C
2 C
3 C*****
4 C*
5 C* PROGRAM TO CONVERT ORDNANCE SURVEY DIGITAL DATA *
6 C* IN CUSTOMER (DMC) FORMAT INTO PACKED BINARY NG *
7 C* COORDINATES STORED AS X(1000), Y(1000), II(1000)*
8 C* WHERE X,Y ARE THE COORDS (DOUBLE PRECISION) AND *
9 C* II IS THE DMB FEATURE CODE ARRAY (HALFWORDS) *
10 C*
11 C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM V1 *
12 C* JUNE 1980
13 C*
14 C* INPUT OF THE DMC CHARACTERS IS ON CHANNEL 5 *
15 C* OUTPUT OF THE BLOCKED BINARY ARRAYS IS CHANNEL 1*
16 C*
17 C*****
18 C
19 REAL*8 X(1000),Y(1000),EAST,RNORTH
20 REAL*8SE,SN,BE,CE,BN,CN,BGS,SQE,SON
21 INTEGER*2 II(1000)
22 COMMON X,Y,II,I
23 CALL STARTR
24 IPEN=0
25 IN=5
26 IOUT=6
27 IKEY=1
28 C
29 C ***** INPUT OF THE DMC RECORDS
30 C
31 5 READ(IN,10,END=9999)IC1,IC2
32 10 FORMAT(2I4)
33 C
34 c ***** test for end of file, start of sheet,
35 c ***** start of feature and entry into new
36 c ***** basic grid square respectively
37 C
38 IF(IC1.EQ.-3)GOTO9999
39 IF(IC1.EQ.-1)GOTO1000
40 IF(IC1.EQ.-4)GOTO4000
41 IF(IC1.EQ.-9)GOTO9000
42 IF(IC1.LT.0)GOTO5
43 C
44 c ***** compute full ng coord values from the
45 c ***** sheet origin
46 C
47 BE=SQE*BGS
```

```
48      BN=SQN*BGS
49      CE=BGS*FLOAT(IC1)/1000.
50      CN=BGS*FLOAT(IC2)/1000.
51      EAST=SE+BE+CE
52      RNORTH=SN+BN+CN
53      C
54      c ***** test to see if the binary buffer of
55      c ***** output is full
56      C
57      CALL OVERFL(EAST,RNORTH,ICODE)
58      ICODE=ICODE2
59      GOTO5
60      C
61      c *** read in sheet origin, basic grid
62      c *** unit and scale respectively
63      C
64      1000 READ(IN,1010)SE,SN,BGS,SCALE
65      1010 FORMAT(F8.0/F8.0/F8.0/F8.0)
66      GOTO5
67      C
68      C ***** START OF NEW FEATURE
69      C
70      I34
71      C
72      C ***** START OF NEW BASIC GRID SQUARE
73      C
74      4000 READ(IN,10)IDUMMY,ICODE
75      ICODE2=ICODE
76      ICODE=-ICODE
77      IKEY=IKEY+1
78      IPEN=1
79      GOTO5
80      9000 READ(IN,9010)SQE, SQN
81      9010 FORMAT(2F4.0)
82      GOTO5
83      C
84      C ***** FLUSH THE BINARY BUFFER BEFORE STOPPING
85      C
86      9999 WRITE(1)X,Y,II
87      STOP 222
88      END
89      C
90      C
91      C
92      SUBROUTINE OVERFL(XNEXT,YNEXT,IPNEXT)
93      C
94      C ***** SUBROUTINE OVERFLOW (TAA)
95      c *** a subroutine to test if the 1000
96      c *** element buffer is full of ng
97      c *** coords. if the buffer still has
98      c *** space in it then return. if the
99      c *** buffer is full then write the
```

```
100      c *** contents to backing store and
101      c *** initialise all pointers
102      C
103          REAL*8 X(1000),Y(1000),EAST,RNORTH
104          REAL*8 XNEXT,YNEXT
105          INTEGER*2 II(1000)
106          COMMON X,Y,II,I
107          I=I+1
108          IF(I.EQ.1001)GOTO10
109      5  X(I)=XNEXT
110          Y(I)=YNEXT
111          II(I)=IPNEXT
112          RETURN
113      10 I=1
114          WRITE(1)X,Y,II
115          DOLLK=1,1000
116          X(K)=0.
117          Y(K)=0.
118      11 II(K)=0
119          GOTO5
120          END
121      C
122      C
123      C
124          SUBROUTINE STARTR
125      C
126      C ***** SUBROUTINE STARTER (TAA)
127      c *** a subroutine to create the 1000
128      c *** element buffer --- set all
129      c *** pointers to zero and empty the
130      c *** buffer
131      C
132          REAL*8X(1000),Y(1000),EAST,RNORTH
133          INTEGER*2 II(1000)
134          COMMON X,Y,II,I
135          DOLI=1,1000
136          X(I)=0.
137          Y(I)=0.
138      1  II(I)=0
139          I=0
140          RETURN
141          END
END OF FILE
```

ANNEX 5b

FORTTRAN source listing for program TO.RASTER (see section 5.3.4)

```
1      C***** T H I S   I S   T O . R A S T E R *****
2      C
3      C
4      C*****
4.8    C*
5.6    C* PROGRAM TO PERFORM A VECTOR TO RASTER
6.4    C* CONVERSION ON ORDNANCE SURVEY DIGITAL DATA
7.2    C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM VIA
8      C* JULY 1980
8.8    C* PROGRAM TAKES AS INPUT ON CHANNEL 1 UNFORMATTED
9.6    C* BINARY STRINGS OF NG COORDINATES IN BLOCKS OF
10.4   C* 1000 ELEMENTS. THESE BLOCKS MUST BE STORED AS
11.2   C* REAL*8 ARRAYS IN THE FORM X(1000), Y(1000) AND
12     C* AN INTEGER*2 II(1000) ARRAY WHICH REPRESENTS THE*
12.8   C* DMB F/CODES. THESE CODES ARE ALWAYS POSITIVE
13.6   C* UNLESS THE POINT IS THE FIRST IN A NEW FEATURE
14.4   C* TYPE, IN WHICH CASE IT IS NEGATED.
15.2   C*
16     C* OUTPUT IS TO CHANNEL 2 IN THE FORM:
16.8   C* RASTER(5000,200) WHICH IS STORED IN UNFORMATTED
17.6   C* BINARY FOR EFFICIENT DATA TRANSFER
18.4   C*
19.2   C*****
21     C
22     C
23     C
24     C
25     LOGICAL*1 RASTER(5000,200)
26     INTEGER*2 ICODE(1000)
27     INTEGER*2 INTEG
28     LOGICAL*1 LOGIC(2)
29     LOGICAL*1 FF/ZFF/
30     REAL*8 SX(1000),SY(1000)
31     EQUIVALENCE(INTEG,LOGIC(1))
32     COMMON/IO/SX,SY,ICODE,II,XORIG,YORIG
33     COMMON/BUFF/RASTER
34     C
35     C-----INITIALISATION
36     C
41     XORIG=426500.
42     YORIG=542500.
43     KK=0
44     PLTSCL=1250
45     PIXEL=0.1
46     TLRNCE=PIXEL/2.
47     ISTRIP=200
48     JSTRIP=5000
```

```
49      WRITE(6,4)
50      4 FORMAT(' GIVE THE STRIP NUMBER (I2)')
51      READ(5,3)NSTRIP
52      WRITE(6,71)NSTRIP
53      71 FORMAT(I5)
54      3 FORMAT(I2)
55      LOWY=NSTRIP*ISTRIP
56      IHIY=LOWY+ISTRIP
57      INTEG=0
58      DO 5 I=1,JSTRIP
59      DO 5 J=1,ISTRIP
60      5 RASTER(I,J)=LOGIC(2)
61      WRITE(6,1)
62      1 FORMAT(' INITIALISATION DONE')
63      CALL STARTR
64      C
65      C-----INPUT OF FIRST COORDINATES
66      C
67      CALL INPUT(X,Y,L)
68      10 KK=KK+1
69      X1=X
70      Y1=Y
71      L1=L
72      C
73      C-----INPUT OF NEXT COORDINATES
74      C
75      CALL INPUT(X,Y,L)
76      IF(L.LT.0)GOTO11
77      GOTO20
78      11 IF(L1.GE.0)GOTO10
79      C
80      C-----HANDLING OF PT SYMBOL
81      C
82      IX=INT(X1/PIXEL)+1
83      IY=INT(Y1/PIXEL)+1
84      IF(IY.LE.LOWY.OR.IY.GT.IHIY)GOTO10
85      IY=IY-LOWY
86      INTEG=IABS(L1)
87      RASTER(IX,IY)=LOGIC(2)
88      GOTO10
89      C
90      C-----DETERMINATION OF ACTUAL MAX AND MIN X AND Y
90.2    C-----FOR TOLERANCE TESTS LATER
91      C
92      20 CALL MAXMIN(X1,Y1,X,Y,XMA,XMI,YMA,YMI)
93      C
94      C-----TO ORDER END PTS IN X FOR THE DO LOOP
95      C
96      IF(X-X1)50,55,60
97      50 XMAX=X1
98      XMIN=X
99      YMAX=Y1
```



```
100          YMIN=Y
101          GOTO70
102          C
103          C-----TO HANDLE PERFECTLY VERTICAL LINES
104          C          (N-S RUNNING)
105          C-----FIRST CHECK IF POINT IS COINCIDENT WITH
105.2        C-----LAST ONE (ZERO SEGMENT LENGTH)
106          C
107          55 IF(Y-Y1)56,10,57
108          56 ISTART=IFIX(Y/PIXEL)
109          IEND=IFIX(Y1/PIXEL)
110          GOTO58
111          57 ISTART=IFIX(Y1/PIXEL)
112          IEND=IFIX(Y/PIXEL)
113          58 IX=INT(X/PIXEL)+1
114          DO 59 I=ISTART,IEND
115          IY=I+1
116          IF(IY.LE.LOWY.OR.IY.GT.IHIY)GOTO59
117          IY=IY-LOWY
118          INTEG=IABS(L)
119          RASTER(IX,IY)=LOGIC(2)
120          59 CONTINUE
121          GOTO 10
122          60 XMAX=X
123          XMIN=X1
124          YMAX=Y
125          YMIN=Y1
126          C
127          C-----SOLUTION FOR Y'S GIVEN X
128          C
129          70 ISTART=IFIX(XMIN/PIXEL)
130          IEND=IFIX(XMAX/PIXEL)
131          DO 100 I=ISTART,IEND
132          XI=I*PIXEL
133          YGEN=((YMAX-YMIN)*((XI-XMIN)/(XMAX-XMIN)))+YMIN
134          C
135          C-----TOLERANCE TEST FOR PTS NEAR LINE ENDS
136          C
137          IF(YGEN.LT.YMI)GOTO80
138          IF(YGEN.GT.YMA)GOTO85
139          GOTO90
140          80 RLIMIT=YMIN-YGEN
141          IF(RLIMIT.GT.TLRNCE)GOTO100
142          GOTO90
143          85 RLIMIT=YGEN-YMAX
144          IF(RLIMIT.GT.TLRNCE)GOTO100
145          90 CONTINUE
146          C
147          C-----ASSIGNMENT OF THE DESIRED PIXEL SQUARE
148          C
149          IX=I+1
150          IY=INT(YGEN/PIXEL)+1
```

```
151         IF(IY.LE.LOWY.OR.IY.GT.IHIY)GOTO100
152         IY=IY-LOWY
153         INTEG=IABS(L)
154         RASTER(IX,IY)=LOGIC(2)
155     100 CONTINUE
156     C
157     C-----SOLUTION OF X'S GIVEN Y
158     C
159     C-----ORDERING OF END PTS IN Y FOR DO LOOP
160     C
161         103 IF(Y-Y1)105,10,115
162         105 XMAX=X1
163             XMIN=X
164             YMAX=Y1
165             YMIN=Y
166             GOTO120
167         115 XMAX=X
168             XMIN=X1
169             YMAX=Y
170             YMIN=Y1
171         120 ISTART=IFIX(YMIN/PIXEL)
172             IEND=IFIX(YMAX/PIXEL)
173             DO 200 I=ISTART,IEND
174                 YI=I*PIXEL
175                 XGEN=((XMAX-XMIN)*((YI-YMIN)/(YMAX-YMIN)))+XMIN
176     C
177     C-----TOLERANCE TESTS FOR PTS NEAR THE LINE ENDS
178     C
179             IF(XGEN.LT.XMI)GOTO130
180             IF(XGEN.GT.XMA)GOTO140
181             GOTO150
182         130 RLIMIT=XMIN-XGEN
183             IF(RLIMIT.GT.TLRNCE)GOTO200
184             GOTO150
185         140 RLIMIT=XGEN-XMAX
186             IF(RLIMIT.GT.TLRNCE)GOTO200
187         150 CONTINUE
188     C
189     C-----ASSIGNMENT OF THE DESIRED PIXEL SQUARE
190     C
191             IY=I+1
192             IX=INT(XGEN/PIXEL)+1
193             IF(IY.LE.LOWY.OR.IY.GT.IHIY)GOTO200
194             IY=IY-LOWY
195             INTEG=IABS(L)
196             RASTER(IX,IY)=LOGIC(2)
197         200 CONTINUE
198             GOTO10
199             END
200     C
201     C
202     C
```

```
203      C
204      C
205      C
206      SUBROUTINE INPUT(CURRX,CURRY,LCURR)
206.02   C
206.05   C-----A SUBROUTINE TO INTERFACE THE INPUT VECTOR
206.1    C-----BUFFER OF 1000 ELEMENTS WITH THE MAIN CODE
206.15   C-----THE ROUTINE RETURNS THE CURRENT VALUES OF
206.2    C-----X,Y AND F/CODE (LCURR) FROM THE BUFFER
206.25   C-----IT HANDLES INSTANCES WHEN THE BUFFER IS
206.3    C-----EMPTY AND SO ON. THE NG COORDS PASSED TO
206.35   C-----THE MAIN CODE HAVE HAD THE MAP SHEET
206.4    C-----ORIGIN SUBTRACTED FROM THEM TO REMOVE
206.45   C-----THE NEED FOR REAL*8 PRECISION
206.5    C
207      REAL*8 SX(1000),SY(1000)
208      INTEGER*2 ICODE(1000)
209      COMMON/IO/SX,SY,ICODE,II,XORIG,YORIG
210      IF(II.EQ.1001)CALL STARTR
211      CURRX=SX(II)-XORIG
212      CURRY=SY(II)-YORIG
213      LCURR=ICODE(II)
214      II=II+1
215      IF(CURRX.LT.0.)CALL PEXIT
216      RETURN
217      END
218      C
219      C
220      C
221      C
222      C
223      C
224      SUBROUTINE STARTR
224.1    C
224.2    C-----A SUBROUTINE TO READ IN THE BUFFER OF 1000
224.3    C-----NG COORDS AND F/CODES WHEN NECESSARY. THE
224.4    C-----POINTERS ARE SET BACK TO 1 TOO.
224.5    C
225      REAL*8 SX(1000),SY(1000)
226      INTEGER*2 ICODE(1000)
227      COMMON/IO/SX,SY,ICODE,II,XORIG,YORIG
228      IN=1
229      READ(IN)SX,SY,ICODE
230      II=1
231      RETURN
232      END
233      C
234      C
235      C
236      C
237      C
238      C
```

```
239          SUBROUTINE MAXMIN(X1,Y1,X2,Y2,XMAX,XMIN,
239.01      1YMAX,YMIN)
239.1      C
239.2      C-----A SUBROUTINE TO RETURN THE MAX AND MIN
239.3      C-----VALUES OF X AND Y FOR A PAIR OF COORDS
239.4      C-----DESCRIBED BY X1,Y1 AND X2,Y2
239.5      C
240          XMAX=X1
241          XMIN=X1
242          YMAX=Y1
243          YMIN=Y1
244          IF(X1-X2)1,2,2
245      1 XMAX=X2
246          GOTO3
247      2 XMIN=X2
248      3 IF(Y1-Y2)4,5,5
249      4 YMAX=Y2
250          RETURN
251      5 YMIN=Y2
252          RETURN
253          END
254      C
255      C
256      C
257      C
258      C
259      C
260          SUBROUTINE PEXIT
260.1      C
260.2      C-----A SUBROUTINE TO WRITE OUT THE CONTENTS OF
260.3      C-----THE RASTER PARTITION AND STOP THE PROG.
260.4      C
261          LOGICAL*1 RASTER(5000,200)
262          COMMON/BUFF/RASTER
263          WRITE(6,1)
264      1 FORMAT(' BUFFER WRITE BEGINS')
265          WRITE(2)RASTER
266          WRITE(6,2)
267      2 FORMAT(' BUFFER WRITE ENDS')
268          STOP 999
269          END
270
END OF FILE
```

ANNEX 5c

FORTRAN source listing of three utilities found to be very useful in developing raster research on a serial processor. They are LPDRAW, RASTER.SCAN and RASTER% (see section 5.3.5)

```
1 C***** T H I S I S L P D R A W
2 C
3 C*****
4 C*
5 C* A UTILITY PROGRAM TO TAKE A RASTER BUFFER AND *
6 C* GENERATE A CRUDE PLOT OF THE IMAGE ON A LINE *
7 C* PRINTER DEVICE WITH A CHARACTER LENGTH OF UP *
8 C* TO 133. *
9 C*
10 C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM *
11 C* JUNE 1980 *
12 C*
13 C* UP TO 20 DMB FEATURE TYPES CAN BE PLOTTED *
14 C* WHICH ARE GIVEN AS INPUT PARAMETERS ON CHANNEL *
15 C* 5. FORTRAN UNIT 6 IS A GENERAL CHANNEL FOR *
16 C* PROGRAM GENERATED MESSAGES *
17 C*
18 C* THE RASTER DATA ARE READ IN ON CHANNEL 1 AS *
19 C* ONE-BYTE PER ELEMENT OF PARTITIONS WHICH CAN *
20 C* BE SET BY THE USER *
21 C*
22 C* THE RESULTANT PLOT IS OUTPUT ON CHANNEL 2 *
23 C*
24 C*****
25 C
26 INTEGER*2 ITYPE(20), RAST
27 LOGICAL*1 RASTER(5000,200)
28 INTEGER*2 XLINE(100), BLANK, HASH
29 LOGICAL*1 FF/ZFF/
30 LOGICAL*1 LOGIC(2)
31 INTEGER*2 INTEG
32 EQUIVALENCE(INTEG, LOGIC(1))
33 C
34 C *** SET THE CHARACTER TYPE FOR BLANK OR DOT
35 C *** ON THE RESULTANT IMAGE
36 C
37 DATA BLANK/' '/, HASH/'#'/
38 C
39 C *** SET THE DIMENSIONS OF THE RASTER BUFFER
40 C *** PARTITION AND THE NUMBER OF CHARACTERS
41 C *** TO EACH LINE-PRINTER SCAN LINE
```

```
42      C
43          ISTRIP=200
44          JSTRIP=5000
45          LPCHAR=100
46      C
47      C *** COMPUTE SCALING FACTORS
48      C
49          JFACT=JSTRIP/LPCHAR
50          FACT=FLOAT(JSTRIP)/FLOAT(LPCHAR)
51      C
52      C *** TEST IF THE SCALING FACTOR IS AN
53      C *** WHOLE INTEGER...IF NOT REPORT
54      C *** THE CHANCES OF PLOT CORRUPTION
55      C
56          FTEST=FACT-INT(FACT)
57          IF(FTEST.NE.0)WRITE(6,51)
58          51 FORMAT(' *** PLOT MAY BE IN ERROR')
59          KOUNT=0
60          IFACT=ISTRIP/JFACT
61      C
62      C *** READ IN THE USER DEFINED FEATURE TYPES
63      C
64          100 WRITE(6,101)
65          101 FORMAT(' HOW MANY FEATURE TYPES ? (I2)')
66          READ(5,105)NTYPE
67          105 FORMAT(I2)
68          IF(NTYPE.LE.0.OR.NTYPE.GT.20)GOTO100
69          WRITE(6,110)
70          110 FORMAT(' GIVE EACH FEATURE TYPE (I3)')
71          DO 115 I=1,NTYPE
72          115 READ(5,120)ITYPE(I)
73          120 FORMAT(I3)
74          WRITE(2,1)
75          1 FORMAT(1H1)
76      C
77      C *** READ IN THE RASTER PARTITION
78      C
79          5 READ(1,END=99)RASTER
80          WRITE(6,6)KOUNT
81          6 FORMAT(' IN STRIP ',I3)
82          KOUNT=KOUNT+1
83      C
84      C *** SEARCH EACH PIXEL IN A LOCAL CLUSTER
85      C *** SIZE OF WHICH IS GIVEN BY THE SCALING
86      C *** FACTOR TO SEE IF ANY ARE SET WITH A
87      C *** F/CODE AS DECLARED BY THE USER (ITYPE)
88      C
89      C *** WHEN PIXELS ARE FOUND THE CORRESPONDING
90      C *** ELEMENT IN THE LP IMAGE IS SET
91      C
92          ISTART=1-JFACT
93          IEND=0
```

```
94          DO 50 K=1,IFACT
95          DO 10 I=1,100
96      10  XLINE(I)=BLANK
97          ISTART=ISTART+JFACT
98          IEND=IEND+JFACT
99          DO 20 I=1,JSTRIP
100         DO 20 J=ISTART,IEND
101         IX=((I-1)/JFACT)+1
102         DO 15 L=1,NTYPE
103         INTEG=0
104         LOGIC(2)=RASTER(I,J)
105         IF(INTEG.GT.127)LOGIC(1)=FF
106         RAST=INTEG
107     15  IF(RAST.EQ.ITYPE(L))XLINE(IX)=HASH
108     20  CONTINUE
109     C
110     C *** OUTPUT THE MUCH REDUCED RASTER IMAGE
111     C *** ON THE LP -- A SCAN LINE AT A TIME
112     C
113         WRITE(2,30)(XLINE(I),I=1,LPCHAR)
114     30  FORMAT(1X,100A1)
115     50  CONTINUE
116     C
117     C *** GO AND READ IN THE NEXT RASTER PARTITION
118     C
119         GOTO5
120     99  WRITE(2,1)
121         STOP
122         END
END OF FILE
```

```
1 C***** T H I S I S R A S T E R . S C A N
2 C
3 C*****
3.8 C*
4.6 C* PROGRAM TO PRINT-OUT THE ADDRESS LOCATION OF *
5.4 C* EACH NON-ZERO ELEMENT IN A ONE-BYTE RASTER *
6.2 C* BUFFER PARTITIONED INTO 5000 X 200 STRIPS *
7 C*
7.8 C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM *
8.6 C* JUNE 1980 *
9.4 C*
10.2 C* INPUT OF BINARY RASTER PARTITION IS ON CHAN.1 *
11 C* OUTPUT OF RESULTS IS ON CHANNEL 6 *
11.8 C*
12.6 C*****
16 C
17 C
18 LOGICAL*1 RASTER(5000,200)
19 LOGICAL*1 LOGIC(2),FF/ZFF/
20 INTEGER*2 RAST,INTEG
21 EQUIVALENCE(INTEG,LOGIC(1))
22 KOUNT=-1
23 C
24 C ***** READ IN RASTER PARTITION
25 C
26 1 READ(2,END=99) RASTER
27 KOUNT=KOUNT+1
28 WRITE(6,2)KOUNT
29 2 FORMAT(' ***** IN STRIP ',I3,' *****')
30 C
31 C ***** SEARCH EACH PIXEL FOR NON-ZERO VALUES
32 C
33 DO 10 I=1,5000
34 DO 10 J=1,200
35 INTEG=0
36 LOGIC(2)=RASTER(I,J)
37 IF(INTEG.GT.127)LOGIC(1)=FF
38 IF(INTEG.EQ.0)GOTO10
39 C
40 C ***** OUTPUT PIXEL ADDRESSES FOR VALUES >0
41 C
42 WRITE(6,5)I,J
43 5 FORMAT(' ADDRESS ',I5,', ',I5)
44 10 CONTINUE
45 C
46 C ***** GO AND READ NEXT PARTITION
47 C
48 GOTO1
49 99 STOP 111
50 END
END OF FILE
```



```
1 C***** T H I S I S R A S T E R %
2 C
3 C*****
4 C*
5 C* A PROGRAM TO PERFORM SOME ELEMENTARY ANALYSES
6 C* ON A RASTER IMAGE FOR ORDNANCE SURVEY DIGITAL
7 C* DATA
8 C*
9 C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM
10 C* SEPTEMBER 1980
11 C*
12 C* PROGRAM READS IN ONE-BYTE PER ELEMENT RASTER
13 C* PARTITION AND COUNTS SET PIXELS OF EACH DMB
14 C* FEATURE TYPE. THESE COUNTS, TOGETHER WITH
15 C* PERCENTAGES ARE PRESENTED IN TABULAR FORM
16 C*
17 C* INPUT OF THE RASTER PARTITIONS IS ON CHANNEL 1
18 C* OUTPUT OF THE TABULAR RESULTS IS ON CHANNEL 2
19 C* AN UPDATE OF THE JOB'S PROGRESS IS GIVEN ON
20 C* CHANNEL 6
21 C*
22 C*****
24 LOGICAL*1 RASTER(5000,200),LOGIC(2),FF/ZFF/
25 INTEGER*4 PERCNT(130)/130*0/
26 INTEGER*2 INTEG
27 EQUIVALENCE(INTEG,LOGIC(1))
28 C
29 C *** SET THE DIMENSIONS OF THE PARTITION
30 C
31 IXAXIS=5000
32 ISTRIP=200
33 KOUNT=0
34 C
35 C *** INPUT THE RASTER PARTITION
36 C
37 1 READ(1,END=99)RASTER
38 WRITE(6,5)KOUNT
39 C
40 C *** GIVE THE PROGRESS OF THE JOB
41 C
42 5 FORMAT(' ***** IN STRIP ',I2)
43 KOUNT=KOUNT+1
44 C
45 C *** FOR EACH PIXEL UNPACK IT INTO 2 BYTES
46 C *** LOOK AT THE FEATURE CODE OR OTHERWISE
48 C *** THE FOLLOWING VARIABLES STORE THESE
49 C *** ENTITIES:
50 C *** ITOTAL - COUNTS EACH PIXEL
51 C *** ISET - COUNTS SET PIXELS
52 C *** PSET - PERCENTAGE SET PIXELS
53 C *** PERCNT - AN ARRAY OF PERCENTAGE SET
54 C PIXELS -- 1 FOR EACH CODE
```

```
55 C *** PZERO - PERCENTAGE ZERO ELEMENTS
56 C
57 DO 50 J=1, ISTRIP
58 DO 50 I=1, IXAXIS
59 INTEG=0
60 LOGIC(2)=RASTER(I,J)
61 IF(INTEG.GT.127)LOGIC(1)=FF
62 L=INTEG+1
63 50 PERCNT(L)=PERCNT(L)+1
64 GOTO1
65 99 IACCUM=ISTRIP*IXAXIS*KOUNT
66 ISET=0
67 DO 100 I=2,110
68 100 ISET=ISET+PERCNT(I)
69 ITOTAL=ISET+PERCNT(1)
70 C
71 C *** WRITE OUT HEADINGS FOR EACH COLUMN IN TABLE
72 C
73 WRITE(2,110)
74 110 FORAM(///,' BREAKDOWN OF THE RASTER BUFFER',/
75 1//' F/CODE PIXEL COUNT PERCENTAGE OF '
76 2' PERCENTAGE OF'/28X,'SET PIXELS TOTAL '
77 3'BUFFER',/1X,55('-')/)
78 PTOT=(FLOAT(PERCNT(1))/FLOAT(ITOTAL))*100.
79 WRITE(2,101)PERCNT(1),PTOT
80 C
81 C *** WRITE OUT THE RESULTS FOR EACH F/CODE
82 C
83 101 FORMAT(' UNSET',5X,I8,22X,F7.3)
84 DO 120 I=2,110
85 L=I-1
86 PSET=(FLOAT(PERCNT(I))/FLOAT(ISET))*100.
87 PTOT=(FLOAT(PERCNT(I))/FLOAT(ITOTAL))*100.
88 WRITE(2,115)L,PERCNT(I),PSET,PTOT
89 115 FORMAT(4X,I3,5X,I8,7X,F7.3,8X,F7.3)
90 120 CONTINUE
91 C
92 C *** WORK OUT GLOBAL RESULTS FOR WHOLE BUFFER
93 C *** AND WRITE OUT RESULTS AT BOTTOM OF TABLE
94 C
95 PSET=(FLOAT(ISET)/FLOAT(ITOTAL))*100.
96 PZERO=(FLOAT(PERCNT(1))/FLOAT(ITOTAL))*100.
97 WRITE(2,150)ITOTAL,ISET,PSET,PERCNT(1),PZERO,
98 IACCUM
99 150 FORMAT(//' BUFFER SUMMARY:'//' TOTAL PIXELS '
100 1' IN BUFFER = ',I10//' TOTAL SET PIXELS IN ',
101 2' BUFFER = ',I10,' (',F8.4,')'//' TOTAL BLANK'
102 3' PIXELS IN BUFFER = ',I10,' (',F8.4,')'//
103 4' TOTAL PIXELS IN BUFFER (CHECK) ',I10//)
104 STOP
105 END
```

END OF FILE

ANNEX 5d

FORTTRAN source listing of two subroutines to handle
the compaction and uncompactation of OS digital data
in a raster format: (see section 5.4.1)

CODER compacts a raster partition into coded-deltas

BUFF uncompactes the coded-delta data back into raster

```
1          SUBROUTINE CODER(IOUT)
2          C
3          C*****
4          C*
5          C* THIS IS A SUBROUTINE TO RUN-LENGTH-ENCODE A
6          C* RASTER PARTITION AND PRODUCE A 1-BYTE ELEMENT
7          C* COMPACTED EQUIVALENT. THE LENGTHS OF ZERO-CODED
8          C* ELEMENTS ARE STORED AS NEGATIVE STRINGS IN
9          C* 1-BYTE ELEMENTS
10         C*
11         C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM V3
12         C* NOVEMBER 1980
13         C*
14         C* THE RASTER PARTITION IS STORED IN RASTER
15         C* THE COMPRESSED SCAN-LINE IS STORED IN LINE
16         C* EACH COMPRESSED SCAN LINE IS TRANSFERRED TO
17         C* BACKING STORE ON CHANNEL SPECIFIED BY IOUT
18         C*
19         C*****
20         C
21         LOGICAL*1 RASTER(5000,200),LHOLD(2)
22         LOGICAL*1 LOGIC(2),LINE(5000),FLAG
23         LOGICAL*1 FF/ZFF/
24         INTEGER*2 INTEG,IHOLD,KOUNT
25         COMMON/BUFF/RASTER
26         EQUIVALENCE(INTEG,LOGIC(1)),(IHOLD,LHOLD(1))
27         C
28         C *** DECLARE DIMENSION OF RASTER PARTITION
29         C
30         ISTRIP=200
31         JSTRIP=5000
32         NBYTE=0
33         DO 300 J=1,ISTRIP
34         NZERO=0
35         KOUNT=1
36         C
37         C *** FLAG IS FALSE WHEN NO RUN-LENGTH PRECEDES
38         C *** THIS ELEMENT AND TRUE WHILST A R-L IS
```

```
39      C *** BUILDING-UP
40      C
41      C      FLAG=.FALSE.
42      C
43      C *** FOR EACH SCAN-LINE UNPACK THE 1-BYTE
44      C *** ELEMENTS INTO 2-BYTES FOR ARITHMETIC
45      C
46      C      DO 90 I=1,JSTRIP
47      C      INTEG=0
48      C      LOGIC(2)=RASTER(I,J)
49      C      IF(INTEG.GT.127)LOGIC(1)=FF
50      C
51      C *** TEST IF ZERO ELEMENT
52      C *** IF NOT WRITE IT OUT IN THE COMPACTION
53      C *** BUFFER 'LINE'
54      C
55      C      IF(INTEG.EQ.0)GOTO50
56      C
57      C *** IF THERE HAS BEEN A R-L BUILDING UP
58      C *** GO TO LINE 30
59      C
60      C      IF(FLAG)GOTO30
61      C      25 LINE(KOUNT)=LOGIC(2)
62      C      KOUNT=KOUNT+1
63      C
64      C *** GET NEXT ELEMENT FROM THE SCAN LINE
65      C
66      C      GOTO90
67      C
68      C *** IF THE CURRENT PIXEL IS SET, THEN
69      C *** WRITE OUT THE R-L WHICH HAS BUILT
70      C *** UP AS A NEGATIVE NUMBER THEN
71      C *** WRITE OUT THE SET PIXEL
72      C
73      C      30 NZERO=NZERO-127
74      C      IF(NZERO-0)40,35,35
75      C      35 IHOLD=-127
76      C      LINE(KOUNT)=LHOLD(2)
77      C      KOUNT=KOUNT+1
78      C      GOTO30
79      C      40 IF(NZERO.EQ.-127)GOTO45
80      C      NZERO=NZERO+127
81      C      IHOLD=-NZERO
82      C      LINE(KOUNT)=LHOLD(2)
83      C      KOUNT=KOUNT+1
84      C
85      C *** MAX RUN LENGTH OF 127 HAS BEEN
86      C *** GENERATED...WRITE IT OUT AND
87      C *** START TO BUILD-UP ANOTHER
88      C
89      C      45 FLAG=.FALSE.
90      C      NZERO=0
```

```
91          GOTO25
92          50 NZERO=NZERO+1
93          FLAG=.TRUE.
94          90 CONTINUE
96          C *** WE ARE AT THE END OF THIS SCAN LINE
97          C *** THE R-L CODED DATA ARE IN 'LINE'
98          C *** IF NO MORE R-L'S TO TAG ON TO THE END
99          C *** GO TO 110 AND OUTPUT LINE ON CHANN IOUT
100         C *** IF STILL A R-L TO BE TAGGED THEN
101         C *** COUNT THEM OUT AND THEN OUTPUT LINE
103         IF(.NOT.FLAG)GOTO110
104         95 NZERO=NZERO-127
105         IF(NZERO-0)105,100,100
106         100 IHOLD=-127
107         LINE(KOUNT)=LHOLD(2)
108         KOUNT=KOUNT+1
109         GOTO95
110         105 IF(NZERO.EQ.-127)GOTO110
111         NZERO=NZERO+127
112         IHOLD=-NZERO
113         LINE(KOUNT)=LHOLD(2)
114         KOUNT=KOUNT+1
115         110 FLAG=.FALSE.
116         NZERO=0
117         KOUNT=KOUNT-1
118         C
119         C *** OUTPUT 'KOUNT' NO. OF BYTES FROM
120         C *** ARRAY LINE TO CHANNEL IOUT
121         C
122         CALL WRITE(LINE,KOUNT,0,LNUM,IOUT,&9999)
123         C
124         C *** COUNT UP THE TOTAL NUMBER OF BYTES
125         C *** IN THIS PARTITION
126         C
127         NBYTE=NBYTE+KOUNT
128         300 CONTINUE
129         C
130         C *** INFORM USER ON CHANN. 6 OF NO.
131         C *** OF BYTES TRANSFERRED
132         C
133         WRITE(6,1000)NBYTE
134         1000 FORMAT(' TRANSFERRED',I10,' BYTES')
135         RETURN
136         C
137         C *** TROUBLES HAVE DEVELOPED ON OUTPUT
138         C *** INFORM USER AND STOP
139         C
140         9999 WRITE(6,10000)
141         10000 FORMAT(' ** ERROR DETECTED ON DATA TRANSFER')
142         STOP 111
143         END
```

END OF FILE

```
1          SUBROUTINE BUFF(RASTER,*)
2          C
3          C*****
4          C*
5          C* THIS IS A SUBROUTINE TO UNCOMPACT THE BACKING *
6          C* STORE VERSION OF A COMPRESSED OS DIGITAL RASTER*
7          C* FILE BACK INTO ITS FULL REPRESENTATION *
8          C*
9          C* WRITTEN BY TIM ADAMS - UNIVERSITY OF DURHAM V2 *
10         C* NOVEMBER 1980 *
11         C*
12         C* THE COMPRESSED SCAN LINES ARE READ IN ON CHAN 1*
13         C* A 1-BYTE ELEMENT RASTER PARTITION 'RASTER' IS *
14         C* RETURNED TO THE CALLING PROGRAM *
15         C*
16         C*****
17         C
18         LOGICAL*1 RASTER(5000,200),LINE(5000)
19         LOGICAL*1 LOGIC(2),FF/ZFF/
20         INTEGER*2NBYTE,INTEG
21         EQUIVALENCE(INTEG,LOGIC(1))
22         ISTRIP=200
23         DO 80 J=1,ISTRIP
24         KOUNT=1
25         C
26         C *** READ IN THE SCAN LINE FROM CHAN 1
27         C *** NBYTE IS THE NUMBER OF BYTES READ
28         C
29         CALL READ(LINE,NBYTE,0,LNUM,1,&99)
30         C
31         C *** FOR EACH BYTE IN 'LINE' UNCOMPACT
32         C *** THE RUN-LENGTHS (R-L) INTO ZEROS
33         C
34         DO 70 K=1,NBYTE
35         INTEG=0
36         LOGIC(2)=LINE(K)
37         IF(INTEG.GT.127)GOTO50
38         RASTER(KOUNT,J)=LINE(K)
39         KOUNT=KOUNT+1
40         GOTO70
41         50 LOGIC(1)=FF
42         C
43         C *** CONVERT THE NEGATIVE R-L'S TO POSITIVE
44         C
45         L=-INTEG
46         L=L+KOUNT-1
47         C
48         C *** FILL IN THE R-L'S WITH ZEROS
49         C
50         INTEG=0
```

```
51          DO 60 M=KOUNT,L
52          C
53          C *** PUT THE UNCOMPACTED R-L DATA INTO THE
54          C *** APPROPRIATE PLACE IN RASTER BUFFER
55          C
56          RASTER(M,J)=LOGIC(2)
57          60 CONTINUE
58          KOUNT=L+1
59          70 CONTINUE
60          80 CONTINUE
61          RETURN
62          C
63          C *** ON END OF FILE RETURN 1
64          C
65          99 RETURN1
66          END
END OF FILE
```

ANNEX 6a

FORTTRAN listing of a program to generate a magnetic tape
in a DAP-based format for data transfer between the
computing sites of Durham and Queen Mary College, London.

```
1      C
2      C ***** THIS IS Q M C F O R M A T
3      C
4      C
5      C *** THIS IS A PROGRAM TO FORMAT A TAPE TO EBCDIC
6      C *** CHARACTERS FOR TRANSPORTATION TO AN ICL 2980
7      C
8      C *** LOGICAL UNIT 1 HOLDS THE LEVEL 6 DATA STORED
9      C *** AT NUMAC AS UNFORMATTED FORTRAN (64,64)
10     C *** 1-BYTE ELEMENTS
11     C
12     C *** LOGICAL UNIT 2 IS THE GENERAL PURPOSE INPUT
13     C *** CHANNEL FOR COMMUNICATION BETWEEN USER AND
14     C *** PROGRAM
15     C
16     C *** LOGICAL UNIT 3 IS THE OUTPUT TAPE WHICH IS
17     C *** STANDARD LABELLED FIXED BLOCK (4096,64)
18     C
19     C *** LOGICAL UNITS 4-19 ARE FOR THE INPUT OF
20     C *** THE 16 (4096,256) RASTER PARTITONS FOR
21     C *** THE LEVEL 12 DATA. CHANNEL 4 READS PART 0,
22     C *** CHANNEL 5 READS PART 1...AND SO ON
23     C
24     C *** VARIABLES: RASTER HOLDS LEVEL 12 DATA
25     C N.B. BECAUSE OF LIMITATIONS OF FORTRAN AT
26     C NUMAC, THE ARRAY RASTER HAS TO BE
27     C READ-IN IN TWO PASSES..A AND B
28     C THESE ARE EQUIVALENCED WITH RASTER
29     C
30     C L1 HOLDS THE LEVEL 6 DATA
31     C LINE HOLDS THE EBCDIC DATA FOR O/PUT
32     C
33     C
34     LOGICAL*1 A(4096,128),B(4096,128)
35     INTEGER*2 INTEG
36     INTEGER LINE(64)
37     EQUIVALENCE(INTEG,LOGIC(1))
38     EQUIVALENCE(RASTER(1,1),A(1,1))
39     EQUIVALENCE(RASTER(1,129),B(1,1))
40     COMMON RASTER
41     DATA IZERO/'0'/,IONE/'1'/
42     C
```



```
43 C ----- IYCURRE IS THE CURRENT LEVEL 12 SEGMENT
44 C ----- (INITIALLY SET TO 0)
45 C
46         IYCURRE=0
47         IHIT=0
48 C
49 C ----- IFC IS THE FEATURE CODE UNDER SCRUTINY
50 C
51         READ(2,10)IFC
52         10 FORMAT(I5)
53 C
54 C ----- INPUT OF THE APPROPRIATE LEVEL 6 DATA
55 C
56         READ(1'IFC)L1
57 C
58 C ----- OUTPUT OF DATA AS CHARS IN BLOCK 1
59 C ----- OF FILE ON TAPE
60 C
61         DO 15 J=1,64
62         DO 12 N=1,64
63         12 LINE(N)=IZERO
64         DO 13 I=1,64
65         INTEG=0
66         LOGIC(2)=L1(I,J)
67         IF(INTEG.EQ.0)GOTO13
68         LINE(I)=IONE
69         13 CONTINUE
70         WRITE(3,35)(LINE(N),N=1,64)
71         15 CONTINUE
72 C
73 C ----- RETRIEVAL OF APPROPRIATE LEVEL 12
74 C ----- DATA IN SEQUENTIAL ORDER OF
75 C ----- INCREASING X
76 C
77         DO 100 J=1,64
78         DO 100 I=1,64
79 C
80 C ----- TEST LEVEL 6 DATA TO SEE IF A LEVEL 12
81 C ----- SUB-DIVISION IS NEEDED
82 C
83         INTEG=0
84         LOGIC(2)=L1(I,J)
85         IF(INTEG.EQ.0)GOTO100
86 C
87 C ----- YES ! SUBDIVISION NEEDED
88 C
89         IHIT=IHIT+1
90 C ----- SOLUTION OF MAPPING FUNCTION IN X -->
91 C ----- IXS=START, IXF=FINISH RANGE
92 C
93         IXS=((I-1)*64)+1
94         IXF=64*I
```

```
95      C
96      C ----- DETERMINATION OF CELL POSITION OF FILE #2
97      C ----- EACH CELL = 256 ELEMENTS
98      C
99          ARG=(J-1)/4
100         IYCELL=INT(ARG)+1
101         IF(IYCELL.EQ.IYCURRE)GOTO20
102      C
103      C ----- RETRIEVE FROM LVL 12 FILE APPROPRIATE CELL
104      C
105      C
106         JJJ=IYCELL+3
107      C     WRITE(8,23)JJJ
108      C 23  FORMAT(I3)
109         READ(JJJ)A,B
110         IYCURRE=IYCELL
111      C
112      C ----- DETERMINATION OF INCR 1 OR 2 WITHIN
113      C ----- RETRIEVED CELL
114      C
115         20 INCR=J-((IYCELL-1)*4)
116      C
117      C ----- SOLUTION OF MAPPING FUNCTION IN Y (DOWN)
118      C ----- IYS=START, IYF=FINISH RANGE
119      C
120         IYS=((INCR-1)*64)+1
121         IYF=64*INCR
122      C
123      C ----- OUTPUT OF LEVEL 12 DATA AS CHARS IN NEXT
124      C ----- BLOCK ON CHANNEL 3
125      C
126         DO 40 L=IYS,IYF
127         DO 25 N=1,64
128         25  LINE(N)=IZERO
129         DO 30 M=IXS,IXF
130         INTEG=0
131         LOGIC(2)=RASTER(M,L)
132         IF(INTEG.NE.IFC)GOTO30
133         MM=M-IXS+1
134         LINE(MM)=IONE
135         30  CONTINUE
136         WRITE(3,35)(LINE(N),N=1,64)
137         35  FORMAT(64A1)
138         40  CONTINUE
139         100 CONTINUE
140         STOP
141         END
END OF FILE
```

ANNEX 6b

FORTRAN/DAP-FORTRAN listing of a program to convert EBCDIC characters on a magnetic tape to a compacted binary format for immediate entry into the ICL DAP

```
1      C ***** T H I S   I S   F I L E S / P A C K
2      C
3      C
4          PROGRAM FILES
5      C
6      C ----- THIS IS A PROGRAM TO READ THE FILES FROM
7      C ----- NUMAC TAPE AND RESTRUCTURE IT INTO A DAP
8      C ----- BASED FORMAT IE CONVERT IT FROM CHARACTERS
9      C ----- (8 BIT) FORM TO 1 BIT FORM. 'FILE' CAN
10     C ----- HANDLE 300 DAP PLANES AT ONCE.
11     C ----- 'PACKFILE' IS 8 TIMES REDUCTION OF 'FILE'
12     C
13     CHARACTER*64 FILE(19200),PACKFILE(2400)
14     C
15     C ----- NFC (INTEGER*8 TO AVOID DAP CONVERSION)
16     C ----- IS THE FEATURE CODE NUMBER
17     C
18     INTEGER*8 NFC
19     LOGICAL FIRST
20     COMMON/DATA/FILE
21     COMMON/PARM/NPL
22     EQUIVALENCE (FILE,PACKFILE)
23     C
24     C ----- N IS NO. OF F/C'S TO BE DEALT WITH
25     C
26     READ(5,*)N
27     DO 10 I=1,N
28     FIRST=.TRUE.
29     J1=9
30     C
31     C ----- NFC IS F/C, NPL IS NO. OF PLANES
32     C ----- NEEDED AT LEVEL 12
33     C
34     READ(5,*)NFC,NPL
35     C
36     C ----- NPACK SPLITS THE FILE INTO 300 PLANE CHUNKS
37     C
38     NPACK=(NPL-1)/300
39     LNPL=NPL-NPACK*300
40     C
41     C ----- OPEN THE DESIRED OUTPUT FILE FCXXX
42     C ----- WHERE XXX IS FEATURE CODE NUMBER
```

```
43      C
44      CALL CONNECTFILES(NFC)
45      IF (NPACK.EQ.0) GOTO 100
46      C
47      C
48      C ----- FOR FILES >=300 PLANES
49      C
50      C
51      DO 20 K=1,NPACK
52      NPL=300
53      NF=NPL*64
54      READ(10,1000)(FILE(J),J=1,NF)
55      1000  FORMAT(1A64)
56      CALL PACK
57      NP=NPL*8
58      IF (.NOT.FIRST)J1=1
59      WRITE(11,1000)(PACKFILE(J),J=J1,NP)
60      FIRST=.FALSE.
61      20  CONTINUE
62      C
63      C
64      C ----- FOR FILES <300 PLANES
65      C
66      C
67      100  IF (LNPL.EQ.0) GOTO 30
68      NPL=LNPL
69      NF=NPL*64
70      C
71      C ----- INPUT CHARACTER DATA FILE FROM TAPE
72      C
73      READ(10,1000)(FILE(J),J=1,NF)
74      C
75      C ----- PACK IT
76      C
77      CALL PACK
78      NP=NPL*8
79      IF(.NOT.FIRST)J1=1
80      C
81      C ----- WRITE IT OUT IN DAP FORMAT
82      C
83      WRITE(11,1000)(PACKFILE(J),J=J1,NP)
84      30  CALL ICL9HFCLOSE(10)
85      CALL ICL9HFCLOSE(11)
86      10  CONTINUE
87      STOP
88      END
89      C
90      C
91      C
92      C
93      C
94      C
```

```
95          ENTRY SUBROUTINE PACK
96 C ----- THIS IS A SUBROUTINE TO STRIP OFF THE TOP
97 C ----- 7 BITS OF EACH CHARACTER. IE EACH TOP 7
98 C ----- BIT PLANES
99 C
100          CHARACTER FILE(,,300),C(,)
101          COMMON/DATA/FILE
102          COMMON/PARM/NPL
103          LOGICAL LM(,,2400)
104          EQUIVALENCE (LM,FILE)
105          CALL CONVFSI(NPL,1)
106 C
107 C ----- PLANE BY PLANE CONVERT DATA TO DAP FORMAT
108 C
109          DO 10 I=1,NPL
110             C=FILE(,,I)
111             CALL CONVFM1(C)
112 10        FILE(,,I)=C
113 C
114 C ----- COMPRESS ONLY THE 8TH SUCCESSIVE BIT
115 C ----- PLANES INTO THE TOP END OF 'LM'
116 C
117          DO 20 I=1,NPL
118             LM(,,I)=LM(,,8*I)
119 20        CONTINUE
120          CALL CONVFSI(NPL,1)
121          RETURN
122          END
123 C
124 C
125 C
126 C
127 C
128          VME/B JOB CONTROL LANGUAGE MACRO TO CONNECT GIVEN
129          FILE-NAME TO A FORTRAN CHANNEL NO.
130
131          MACRO NAME IS "CONNECTFILES"
132
133
134          MACRO CONNECTFILES IS (
135          REF INT ?A )
136          MACBEGIN
137          IF ?A GT 9 THEN
138          STRING INFILE= ':HLFA100.FCO'+NUMERIC(?A)
139          STRING OUTFILE= ':.TAPE.FCO'+NUMERIC(?A)
140          ELSE
141          STRING INFILE= ':HLFA100.FCO0'+NUMERIC(?A)
142          STRING OUTFILE= ':.TAPE.FCO0'+NUMERIC(?A)
143          FI
144          AF(VAL INFILE,ICL9LF10)
145          CF(VAL OUTFILE)
146          AF(VAL OUTFILE,ICL9LF11,ACC=W)
```

147 RETURN

148 MACEND

END OF FILE

ANNEX 6c

FORTRAN/DAP-FORTRAN listing of a graphics suite written to transpose raster images between levels in the hierarchical data structure and display them on a Sigma colour raster display device.

```
1      C ***** THIS IS OSMAP
2      C ***** ( R A S T E R - G R A P H I C S )
3      C
4          PROGRAM OSMAP
5      C
6      C ----- THIS IS THE MAINLINE 2980 HOST PROGRAM
7      C ----- FOR THE DAP RASTER GRAPHICS PACKAGE
8      C
9      C ----- DECLARATION OF INTEGER*8 MEANS THAT
10     C ----- NO CONVERSION TO DAP FORMAT IS NECESSARY
11     C
12         INTEGER*8 NFC,SUM6,NFIRST
13         INTEGER INDEX(4096)
14         COMMON/FC/NFC
15     C
16     C ----- FILE IS DECLARED SUITABLY LARGE AND
17     C ----- IS, IN EFFECT, THE DAP TO THIS PROGRAM
18     C
19         CHARACTER*64 FILE(14000)
20         COMMON/DATA/FILE
21         COMMON/IND/INDEX,SUM6,NFIRST
22         EXTERNAL OUTPUT
23     C
24     C ----- THE USE OF AN ERROR TRAP ENABLES A SNEAKY
25     C ----- REVERT BACK FROM THE DAP TO THE 2980
26     C
27     C ----- THE FOLLOWING THREE LINES ARE JUST
28     C ----- VME/B MANDATORY FOR ERROR TRAP CATCHING
29     C
30         CALL ICL9HERRORTRAPDR(OUTPUT,IRES)
31         IF(IRES.NE.0)WRITE(6,1)
32         1 FORMAT(' SETMAP UNSUCCESSFUL')
33     C
34     C ----- ASK THE USER TO REQUEST WHICH F/C HE WISHES
35     C
36         WRITE(6,100)
37     100 FORMAT(' WHICH FEATURE CODE?')
38         READ(5,*)NFC
```

```
39      C
40      C ----- INPUT OF THE INDEX LEVEL 6 FILE
41      C
42          NF=1417*8
43          READ(7,1000)(FILE(I),I=1,NF)
44      1000 FORMAT(1A64)
45      C
46      C ----- OPEN AND LINK UP CHANNEL 10 TO APPROPRIATE
47      C ----- NFC LEVEL 12 (DAP FORMAT) FILE
48      C
49          CALL CONNECTFILE(NFC)
50          WRITE(6,500)NFC
51      500  FORMAT(I15)
52      C
53      C ----- EXAMINE DESIRED LEVEL 6 FILE
54      C ----- AND SHUNT THE POSITIONAL
55      C ----- INDEX TO TOP OF LONG VECTOR
56      C
57          CALL GETNFC
58      C
59      C ----- TEMPORARY INSERTION
60      C
61          SUM6=22
62      C
63      C ----- SUM6 IS THE NO. OF SET LEVEL 6 PIXELS
64      C
65          NF=SUM6*8
66          WRITE(6,1001)NF
67      1001 FORMAT(' NF = ',I10)
68      C
69      C ----- READ IN THE DESIRED LEVEL 12 DATA
70      C
71          READ(10,1000)(FILE(I),I=1,NF)
72      C
73      C ----- SET UP THE SIGMA GRAPHICS TERMINAL
74      C
75          CALL SETMAP
76      C
77      C ----- MOVE INTO THE DAP FOR GRAPHICS PRODUCTION
78      C
79          CALL OSMAPPING
80          STOP
81          END
82      C
83      C
84      C
85      C
86      C
87      C
88      VME/B job control language macro to connect a disk
89      file to a specified FORTRAN channel number
90      C
```



```
91 MACRO CONNECTFILE IS (
92 REF INT ?A )
93 MACBEGIN
94 IF ?A GT 9 THEN
95 STRING INFILE= ':.TAPE.FCO'+NUMERIC(?A)
96 ELSE
97 STRING INFILE= ':.TAPE.FCO0'+NUMERIC(?A)
98 FI
99 AF(VAL INFILE,ICL9LF10)
100 RETURN
101 MACEND
102 C
103 C
104 C
105 C
106 C
107 ENTRY SUBROUTINE GETNFC
108 C
109 C ----- THIS IS A SUBROUTINE TO EXTRACT THE
110 C ----- DESIRED LEVEL 6 DAP PLANE AND
111 C ----- DEVELOP AN INTEGER MATRIX 'INDEX' WHICH
112 C ----- IS THE POSITIONAL ELEMENT INFO. SHUNTED
113 C ----- TO THE TOP OF THE LONG VECTOR
114 C
115 INTEGER SUM6,NFIRST,INDEX(,),TINDEX(,)
116 LOGICAL LEVEL6(,,109),LM(,)
117 COMMON/DATA/LEVEL6
118 COMMON/IND/INDEX,SUM6,NFIRST
119 COMMON/FC/NFC
120 EXTERNAL INTEGER MATRIX FUNCTION IPACK
121 C
122 C ----- PUT JUST OUR DESIRED F/C DAP PLANE INTO LM
123 C
124 LM=LEVEL6(,,NFC)
125 C
126 C ----- DETERMINE HOW MANY PIXELS THERE ARE IN 'LM'
127 C
128 SUM6=SUM(LM)
129 C
130 C ----- SET ALL OF INDEX=1
131 C
132 INDEX=1
133 IS=1
134 C
135 C ----- USE RECURSIVE DOUBLING TO SET UP A LONG
136 C ----- VECTOR OF 4096 ELEMENTS AS 1,2,3,4,.....,4096
137 C
138 DO 10 I=1,12
139 INDEX=INDEX+SHRP(INDEX,IS)
140 10 IS=IS*2
141 TINDEX=0
142 C
```

```
143 C ----- MASK THROUGH ONLY THOSE POSITIONS IN WHICH
144 C ----- THERE ARE SET PIXELS (IE WHERE LM = .TRUE.)
145 C ----- RESULT GOES INTO TEMPORARY INDEX 'TINDEX'
146 C
147 C           TINDEX(LM)=INDEX
148 C
149 C ----- SHUNT THE RELEVANT DATA TO THE TOP OF THE
150 C ----- LONG VECTOR
151 C
152 C           INDEX=IPACK(TINDEX)
153 C           TRACE 1(INDEX)
154 C           RETURN
155 C           END
156 C
157 C
158 C
159 C
160 C
161 C           INTEGER MATRIX FUNCTION IPACK(IM)
162 C
163 C ----- TREATING IM AS A LONG VECTOR, THE MAX.
164 C ----- MATRIX POSITION WILL BE 4096 (I.E. ONLY
165 C ----- NEEDS 12-BITS FOR STORAGE).
166 C
167 C ----- THIS IS A FUNCTION TO CHOP OFF BITS 13-32
168 C ----- FROM THE INTEGER STACK FOR EFFICIENCY IN
169 C ----- DAP STORAGE. FUNCTION ISIGMA IS USED TO
170 C ----- SHUNT THE SET PIXELS TO THE TOP OF THE
171 C ----- LONG VECTOR FOR EASY ACCESS.
172 C
173 C           INTEGER IM(,)
174 C           INTEGER COUNT(,)
175 C           LOGICAL LC(,,32),MOVE(,)
176 C           EQUIVALENCE (COUNT,LC)
177 C           EXTERNAL INTEGER MATRIX FUNCTION ISIGMA
178 C           MOVE=IM.EQ.0
179 C           COUNT=0
180 C           COUNT(.NOT.MOVE)=ISIGMA(IFIX(MOVE))
181 C           L=1
182 C           IPACK=IM
183 C           DO 20 I=1,12
184 C             MOVE=LC(,,33-I)
185 C             MOVE=SHLP(MOVE,L)
186 C             IPACK(MOVE)=SHLP(IPACK,L)
187 C             COUNT(MOVE)=SHLP(COUNT,L)
188 C           20 L=L*2
189 C           RETURN
190 C           END
191 C
192 C
193 C
194 C
```

```
195      C
196          INTEGER MATRIX FUNCTION ISIGMA(X)
197      C
198      C ----- THIS IS A FUNCTION TO SHUNT SET PIXELS IN
199      C ----- INTEGER MATRIX X TO THE TOP OF THE MATRIX.
200      C ----- IT USES THE RECURSIVE DOUBLING THEORY OF
201      C ----- BINARY ARITHMETIC.
202      C
203          INTEGER X(,)
204          ISIGMA=X
205          L=1
206          DO 10 I=1,12
207              ISIGMA=ISIGMA+SHRP(ISIGMA,L)
208      10    L=L*2
209          RETURN
210          END
211      C
212      C
213      C
214      C
215      C
216          SUBROUTINE SETMAP
217      C
218      C ----- THIS IS A SUBROUTINE TO INITIALISE THE SIGMA
219      C ----- COLOUR GRAPHICS TERMINAL FOR DISPLAY OF THE
220      C ----- RASTER DATA FROM THE DAP
221      C
222          CALL START
223          CALL SETBUFSIZE(1024,6)
224          CALL CSIZE(2)
225          CALL DRAWPADS
226          CALL INSTRUCTION
227          RETURN
228          END
229      C
230      C
231      C
232      C
233      C
234          SUBROUTINE DRAWPADS
235      C
236      C ----- THIS IS A SUBROUTINE TO DRAW THREE PADS
237      C ----- ON THE RIGHT HAND SIDE OF THE GRAPHICS
238      C ----- SCREEN TO ACT AS MENU SQUARES FOR DESIRED
239      C ----- RESOLUTION OF PLOT
240      C
241          CALL MOVE(560,490,.FALSE.)
242          CALL CHAR(' GRID 512X512 *.')
243          JST=476
244          CALL CSIZE(1)
245          DO 10 J=1,64
246          JST=JST-1
```



```
299 C
300 C ----- IF INSTRUCTION=999 THEN PREPARE TO STOP
301 C
302 999 IF(INT.EQ.999)RETURN
303     IY=511-IY
304 C
305 C ----- CALL APPROPRIATE DRAW ROUTINES -
306 C ----- DEPENDING ON INSTRUCTION
307 C
308     IF(INT.NE.9)GOTO 10
309     CALL DRAW512
310 C     CALL CONVFSI(INT,3)
311     GOTO 999
312 10 IF(INT.NE.10)GOTO 11
313     IX=IX/32+1
314     IY=IY/32+1
315     CALL DRAW1024(IX,IY)
316 C     CALL CONVFSI(INT,3)
317     GOTO 999
318 11 IF(INT.NE.11)GOTO 999
319     IX=IX/16+1
320     IY=IY/16+1
321     CALL DRAW2048(IX,IY)
322 C     CALL CONVFSI(INT,3)
323     GOTO 999
324     END
325 C
326 C
327 C
328 C
329 C
330     SUBROUTINE GRID512(INDEX,SUM6)
331 C
332 C ----- THIS IS A SUBROUTINE TO SET UP COMMON BLOCK
333 C ----- MAP9 AND SPLIT UP THE DAP PLANES INTO 8 X 8
334 C ----- CHUNKS FOR 512 X 512 RESOLUTION LEVEL 9
335 C
336     LOGICAL LEVEL12(,,1750),MAP9(,,8,8),SECT12(,)
337     LOGICAL TEMP(,)
338     INTEGER INDEX(,),SUM6
339     EXTERNAL LOGICAL MATRIX FUNCTION MASKPLANE
340     COMMON/DATA/LEVEL12
341     COMMON/MAPS9/MAP9
342 C
343 C ----- INITIALISE MAP9
344 C
345     DO 100 I=1,8
346     DO 100 J=1,8
347 100 MAP9(,,I,J)=.FALSE.
348 C
349 C ----- DO FOR EACH BIT PLANE OF THE LEVEL 12 DATA
350 C
```

```
351         DO 10 K=1,SUM6
352     C
353     C ----- SET 'N' TO BE THE POSITIONAL LEVEL 6 POSN.
354     C ----- IN THE MAP-SHEET INDEX
355     C
356         N=INDEX(K)
357     C
358     C ----- COMPUTE CORRESPONDING POSITIONAL COORDS IN
359     C ----- THE LEVEL 12 IMAGE:
360     C         I64,J64 ARE THE x,y COORDS IN LEVEL 6
361     C         I8,J8 ARE THE x,y COORDS OF THE 8 X 8 CHUNK
362     C         IN THE 64 X 64
363     C         I18,J18 ARE THE x,y COORDS WITHIN THE 8 X 8
364     C
365         IGRID=64/8
366         I64=(N/64)+1
367         IF(N.EQ.4096) I64=64
368         I8=((I64-1)/IGRID)+1
369         J64=N-((I64-1)*64)
370         J8=((J64-1)/IGRID)+1
371         JJ8=N-(((I64-1)*64)+(J8-1)*IGRID)
372         I18=((N-(((I8-1)*64*IGRID)+J64))/64)+1
373     C
374     C ----- SET SECT12 TO BE CURRENT LEVEL 12 DAP PLANE
375     C
376         SECT12=LEVEL12(,,K)
377         TRACE 1(SECT12)
378     C
379     C ----- REDUCE TO A LEVEL 9 REPRESENTATION
380     C
381         CALL COMPRESS(3,SECT12,I18,J18)
382         TRACE 1(SECT12)
383     C
384     C ----- PLACE REDUCED IMAGE IN CORRECT POSN. OF THE
385     C ----- 512 X 512 IMAGE
386     C
387         MAP9(MASKPLANE(I18,J18,3),I8,J8)=SECT12
388         TEMP=MAP9(,,I8,J8)
389         TRACE 1(TEMP)
390     10 CONTINUE
391         RETURN
392         END
393     C
394     C
395     C
396     C
397     C
398         SUBROUTINE GRID1024(INDEX,SUM6)
399     C
400     C ----- THIS IS A SUBROUTINE TO SET UP COMMON BLOCK
401     C ----- MAP10 AND SPLIT THE DAP PLANES TO 4x4 CHNKS
402     C ----- FOR A LEVEL 10 REPRESENTATION
```

```
403      C
404          LOGICAL LEVEL6(,,109)
405          LOGICAL LEVEL12(,,1750),MAP10(,,16,16)
406          LOGICAL SECT12(,)
407          INTEGER INDEX(,),SUM6
408          EXTERNAL LOGICAL MATRIX FUNCTION MASKPLANE
409          COMMON/DATA/LEVEL6,LEVEL12
410          COMMON/MAPS10/MAP10
411      C
412      C ----- INITIALISE MAP10
413      C
414          DO 100 I=1,16
415          DO 100 J=1,16
416      100  MAP10(,,I,J)=.FALSE.
417      C
418      C ----- DO FOR EACH BIT-PLANE OF THE LEVEL 12 DATA
419      C
420          DO 10 K=1,SUM6
421      C
422      C ----- SET N TO BE THE POSITIONAL LEVEL 6 POSN.
423      C ----- IN THE MAP-SHEET INDEX
424      C
425          N=INDEX(K)
426      C
427      C ----- COMPUTE CORRESPONDING POSTN. COORDS IN
428      C ----- THE LEVEL 12 DATA:
429      C          I64,J64 IS x,y IN 64 X 64 DAP-PLANE
430      C          I16,J16 IS x,y OF EACH 4x4 CHUNK
431      C          I4,J4 IS x,y WITHIN EACH 4x4 CHUNK
432      C
433      C
434          IGRID=64/16
435          I64=(N/64)+1
436          IF(N.EQ.4096) I64=64
437          I16=((I64-1)/IGRID)+1
438          J64=N-((I64-1)*64)
439          J16=((J64-1)/IGRID)+1
440          J4=N-(((I64-1)*64)+(J16-1)*IGRID)
441      C          I4=((N-(((I16-1)*64*IGRID)+J64))/64)+1
442          L=(I16-1)*64*IGRID+(J16-1)*IGRID
443          I4=4
444          IF(N-L.GT.4) GOTO 20
445          I4=1
446          GOTO 40
447      20  IF(N-L-64.GT.4) GOTO 30
448          I4=2
449          GOTO 40
450      30  IF(N-L-128.GT.4) GOTO 40
451          I4=3
452      40  CONTINUE
453      C
454      C ----- SET SECT12 TO CURRENT L12 PLANE
```

```
455 C
456 C      SECT12=LEVEL12(,,K)
457 C
458 C ----- REDUCE TO A LEVEL 10 REPRESENTATION
459 C
460 C      CALL COMPRESS(4,SECT12,I4,J4)
461 C
462 C ----- PLACE REDUCED IMAGE IN CORRECT LEVEL10
463 C ----- POSITION.
464 C
465 C      MAP10(MASKPLANE(I4,J4,4),I16,J16)=SECT12
466 10  CONTINUE
467     RETURN
468     END
469 C
470 C
471 C
472 C
473 C
474 C      SUBROUTINE GRID2048(INDEX,SUM6)
475 C
476 C ----- THIS IS A SUBROUTINE TO SET UP COMMON BLOCK
477 C ----- MAP11 AND SPLIT UP DAP PLANES TO 2x2 CHUNKS
478 C ----- FOR A 2048 X 2048 LEVEL 11 DISPLAY
479 C
480 C      LOGICAL LEVEL12(,,1750),MAP11(,,32,32)
481 C      LOGICAL SECT12(,)
482 C      INTEGER INDEX(,),SUM6
483 C      EXTERNAL LOGICAL MATRIX FUNCTION MASKPLANE
484 C      COMMON/DATA/LEVEL12
485 C      COMMON/MAPS11/MAP11
486 C
487 C ----- INITIALISE MAP11
488 C
489 C      DO 100 I=1,32
490 C      DO 100 J=1,32
491 100  MAP11(,,I,J)=.FALSE.
492 C
493 C ----- DO FOR EACH BIT PLANE OF LEVEL 12 DATA
494 C
495 C      DO 10 K=1,SUM6
496 C
497 C ----- SET N TO BE THE LEVEL 6 POSITION IN THE
498 C ----- MAP-SHEET INDEX
499 C
500 C      N=INDEX(K)
501 C
502 C ----- COMPUTE CORRESPONDING POSITION COORDS IN THE
503 C -----4096x4096 IMAGE:
504 C      I64,J64 IS x,y IN 64x64
505 C      I32,J32 IS x,y OF THE 2x2 CHUNK INSIDE 64x64
506 C      I2,J2 IS x,y INSIDE THE 2x2
```



```
507 C
508 IGRID=64/32
509 I64=(N/64)+1
510 IF(N.EQ.4096) I64=64
511 I32=((I64-1)/IGRID)+1
512 J64=N-((I64-1)*64)
513 J32=((J64-1)/IGRID)+1
514 J2=N-(((I64-1)*64)+(J32-1)*IGRID)
515 C I2=((N-(((I32-1)*64*IGRID)+J64))/64)+1
516 L=(I32-1)*64*IGRID+(J32-1)*IGRID
517 I2=1
518 IF (N-L.GT.2) I2=2
519 C
520 C ----- SET SECT12 TO BE CURRENT LEVEL 12 DAP PLANE
521 C
522 SECT12=LEVEL12(,,K)
523 C
524 C ----- REDUCE TO A LEVEL 11 REPRESENTATION
525 C
526 CALL COMPRESS(5,SECT12,I2,J2)
527 C
528 C ----- PLACE REDUCED IMAGE IN CORRECT
529 C ----- POSITION OF LEVEL 11
530 C
531 MAP11(MASKPLANE(I2,J2,5),I32,J32)=SECT12
532 10 CONTINUE
533 RETURN
534 END
535 C
536 C
537 C
538 C
539 C
540 SUBROUTINE COMPRESS(P,SECT12,X,Y)
541 C
542 C *** THIS IS A SUBROUTINE TO DIVIDE CONCEPTUALLY
543 C *** THE 64x64 DAP PLANE SECT12 (WHICH IS A
544 C *** LEVEL 12 DAP IMAGE) INTO CHUNKS OF 2**P
545 C *** THEN ON THE BASIS OF THESE CHUNKS IT PERFORMS
546 C *** A 'FOLDING' OPERATION TO REDUCE THE 2**Px2**P
547 C *** UNITS INTO 1 PIXEL (SCATTERED WITHIN THE
548 C *** 64x64 IMAGE. THESE SCATTERED DATA ARE SHUNTED
549 C *** TO THE TOP 2**Px2**P AREA OF THE MATRIX AND
550 C *** MOVED TO THEIR CORRECT POSITION SPECIFIED BY
551 C *** SUPPLIED PARAMETERS X AND Y
552 C
553 C
554 C
555 INTEGER X,Y,P,V(),XX(,),COUNT(,),TCOUNT(,)
556 LOGICAL SECT12(,),MASK(,),LC(,,32),LTC(,,32),
557 IMOVE(,)
558 EQUIVALENCE(COUNT,LC(,,1)),(TCOUNT,LTC(,,1))
```

```
559      C
560      C ----- SET N=NO OF PIXELS IN THE 2**P
561      C ----- CHUNK THEN SUBTRACT 1
562      C
563          M=2**P
564          N=2**(6-P)-1
565      C
566      C ----- SUCCESSIVELY SHIFT NTH AND WEST (FOLDING
567      C ----- OPERATION) PRESERVING ANY SET PIXELS
568      C
569          DO 10 I=1,N
570             SECT12=SHNC(SECT12,1).OR.SECT12
571          10 SECT12=SHWC(SECT12,1).OR.SECT12
572             N=N+1
573      C
574      C ----- SET V (A 64 ELEMENT INT VECTOR) = 1
575      C
576          V=1
577          L=1
578      C
579      C ----- USE RECURSIVE DOUBLING ON THE VECTOR V TO
580      C ----- CREATE A VECTOR OF THE FORM: 1,2,3,4,..64
581      C
582          DO 20 I=1,6
583             V=V+SHRP(V,L)
584          20 L=L*2
585      C
586      C ----- SUBTRACT 1 FROM EVERY VALUE OF VECTOR V
587      C
588          V=V-1
589      C
590      C ----- BUILD XX A MATRIX OF SUCCESSIVE ROWS OF V
591      C
592          XX=MATR(V)
593      C
594      C ----- SET UP A MASK OF .TRUE. ONLY WHERE NO WILL
595      C ----- DIVIDE BY N EXACTLY
596      C
597          MASK=XX-(XX/N)*N.EQ.0
598      C
599      C ----- NOW XX = SUCCESSIVE COLUMNS OV VECTOR V
600      C
601          XX=MATC(V)
602      C
603      C ----- NOW AMEND MASK SO THAT IT IS ONLY .TRUE. IF
604      C ----- THE COLUMNS WILL DIVIDE BY N EXACTLY BUT
605      C ----- PRESERVE THE PROPERTIES OF MASK
606      C
607          MASK=MASK.AND.XX-(XX/N)*N.EQ.0
608      C
609      C ----- SET SECT12 ONLY EQUAL TO ELEMENTS WHERE
610      C ----- MASK IS TRUE
```

```
611      C
612          SECT12=MASK.AND.SECT12
613          TRACE 1(SECT12)
614      C
615      C ----- SET COUNT = INTEGER EQUIV OF .NOT.MASK
616      C
617          COUNT=IFIX(.NOT.MASK)
618          L=1
619      C
620      C ----- RECURSIVE DOUBLING ROUTINE TO SHUNT THE
621      C ----- DISPERSED VALUES TO TOP N*N AREA OF THE
622      C ----- MATRIX
623      C
624      C
625          DO 30 I=1,6
626          COUNT=COUNT+SHSP(COUNT,L)
627      30  L=L*2
628          L=1
629          DO 35 I=1,6
630          MOVE=LC(, , 33-I)
631          SECT12(MOVE)=SHNP(SECT12,L)
632          MASK(MOVE)=SHNP(MASK,L)
633          COUNT(MOVE)=SHNP(COUNT,L)
634      35  L=L*2
635          L=1
636          TRACE 1(SECT12)
637          TCOUNT=IFIX(.NOT.MASK)
638          DO 40 I=1,6
639          TCOUNT=TCOUNT+SHEP(TCOUNT,L)
640      40  L=L*2
641          L=1
642          DO 45 I=1,6
643      C
644          MOVE=LTC(, , 33-I)
645          SECT12(MOVE)=SHWP(SECT12,L)
646          TCOUNT(MOVE)=SHWP(TCOUNT,L)
647      C
648      45  L=L*2
649      C
650          IX=(X-1)*M
651          IY=(Y-1)*M
652          IY=(Y-1)*64/N
653          TRACE 1(SECT12)
654          SECT12=SHEP(SHSP(SECT12,IY),IX)
655          TRACE 1(SECT12)
656          RETURN
657          END
658      C
659      C
660      C
661      C
662      C
```

```
663          LOGICAL MATRIX FUNCTION MASKPLANE(I,J,P)
664          C
665          C ----- THIS IS A FUNCTION TO SET UP A MASK SET TO
666          C ----- .FALSE. EXCEPT IN A 2**P X 2**P AREA IN
667          C ----- POSITION I,J WHICH IS SET TO .TRUE.
668          C
669          INTEGER P
670          LOGICAL X(,),Y(,)
671          ISIZE=2**P
672          ISTART=((I-1)*ISIZE)+1
673          IEND=ISTART+(ISIZE-1)
674          JSTART=((J-1)*ISIZE)+1
675          JEND=JSTART+(ISIZE-1)
676          MASKPLANE=.FALSE.
677          X=COLS(ISTART,IEND)
678          Y=COLS(JSTART,JEND)
679          Y=TRAN(Y)
680          MASKPLANE=X.AND.Y
681          RETURN
682          END
683          C
684          C
685          C
686          C
687          C
688          SUBROUTINE DRAW512
689          C
690          C ----- THIS IS A SUBROUTINE TO OUTPUT A LEVEL 9
691          C ----- 512 X 512 IMAGE TO THE SIGMA
692          C
693          LOGICAL MAP9(,,8,8)
694          COMMON/MAPS9/MAP9
695          CALL DRAW(MAP9,0,511)
696          RETURN
697          END
698          C
699          C
700          C
701          C
702          C
703          SUBROUTINE DRAW1024(I,J)
704          C
705          C ----- THIS IS A SUBROUTINE TO OUTPUT A LEVEL 10
706          C ----- 1024 X 1024 IMAGE TO THE SIGMA
707          C
708          LOGICAL MAP10(,,16,16),OUT(,,8,8)
709          COMMON/MAPS10/MAP10
710          DO 10 K=1,8
711          DO 10 L=1,8
712          10 OUT(,,K,L)=.FALSE.
713          I8=I+7
714          J8=J+7
```

```
715         II=0
716         JJ=0
717         IF(I8.GT.16) I8=16
718         IF(J8.GT.16) J8=16
719         DO 20 K=I,I8
720         II=1+II
721         JJ=0
722         DO 20 L=J,J8
723         JJ=1+JJ
724         OUT(,,II,JJ)=MAP10(,,K,L)
725     20    CONTINUE
726         CALL DRAW(OUT,0,511)
727         RETURN
728         END
729     C
730     C
731     C
732     C
733     C
734         SUBROUTINE DRAW2048(I,J)
735     C
736     C ----- THIS IS A SUBROUTINE TO OUTPUT A LEVEL 11
737     C ----- 2048 X 2048 IMAGE TO THE SIGMA
738     C
739         LOGICAL MAP11(,,32,32),OUT(,,8,8)
740         COMMON/MAPS11/MAP11
741         DO 10 K=1,8
742         DO 10 L=1,8
743     10    OUT(,,K,L)=.FALSE.
744         I8=I+7
745         J8=J+7
746         II=0
747         JJ=0
748         IF(I8.GT.32) I8=32
749         IF(J8.GT.32) J8=32
750         DO 20 K=I,I8
751         II=1+II
752         JJ=0
753         DO 20 L=J,J8
754         JJ=1+JJ
755         OUT(,,II,JJ)=MAP11(,,K,L)
756     20    CONTINUE
757         CALL DRAW(OUT,0,511)
758         RETURN
759         END
760     C
761     C
762     C
763     C
764     C
765         SUBROUTINE DRAW(PICT,IST,JST)
766     C
```

```
767 C ----- THIS IS A SUBROUTINE TO TAKE A 512 X 512
768 C ----- IMAGE CONVERT IT TO SIGMA FORMAT AND
769 C ----- CAUSE AN ERROR TO FORCE THE HOST MACHINE
770 C ----- TO INTERRUPT THE DAP FOR THE I/O
771 C
772 C          LOGICAL PICT(,,8,8)
773 C          INTEGER IG,JG,ISIZE
774 C
775 C ----- SET UP COLOUR OPTIONS
776 C
777 C          CHARACTER*1 IA(,),CHARR()/'1','2','3','4','5',
778 C          1'6','7','8','9','A','B','C','D','E','F','0'/
779 C          COMMON/DISPLAY/IA
780 C          COMMON/DISSIZE/ISIZE,IG,JG
781 C          COMMON/COUNT/INSTR
782 C          DO 10 I=1,8
783 C          IG=IST+(I-1)*64
784 C          DO 10 J=1,8
785 C          INSTR=1
786 C          JG=JST-(J-1)*64
787 C          IA=CHARR(8)
788 C
789 C ----- DISPLAY LINEWORK IN COLOUR OPTION 1
790 C
791 C          IA(PICT(,,J,I))=CHARR(1)
792 C          CALL CONVMF1(IA)
793 C          IF(I.EQ.8.AND.J.EQ.8)INSTR=0
794 C          ERROR 8999
795 C          10 CONTINUE
796 C          RETURN
797 C          END
798 C
799 C
800 C
801 C
802 C
803 C          SUBROUTINE OUTPUT(L,NUM)
804 C
805 C ----- THIS IS A SUBROUTINE IN 2980 FORTRAN
806 C ----- IT'S INVOKED EACH TIME ERROR CONDITION 8999
807 C ----- OCCURS IN THE DAP AND DISPLAYS THE SIGMA
808 C ----- FORMATTED IMAGE IA TO THE COLOUR DISPLAY
809 C
810 C          CHARACTER*1 L,IA(64,64)
811 C          INTEGER*8 ISIZE,IG,JG,INSTR
812 C          INTEGER NUM
813 C          COMMON/COUNT/INSTR
814 C          COMMON/DISSIZE/ISIZE,IG,JG
815 C          COMMON/DISPLAY/IA
816 C
817 C ----- VME/B MANDATORY FOR ERRORTRAP
818 C
```

```
819         IF (L.NE.'S') RETURN
820         IF (NUM.NE.8999) RETURN
821         CALL CONT
822         CALL ICL9HERRORTABLE(0,-1,-1,0,0,0,0,
823         1-1,0,-1,1,IRES)
824     C
825     C ----- DISPLAY THE DATA IN THIS DAP PLANE
826     C
827         JG=JG+1
828         DO 10 J=1,64
829         JG=JG-1
830         IIG=IG
831         JJG=JG
832         CALL MOVE(IIG,JJG,.FALSE.)
833         CALL OUTLINE(IA(1,J),64)
834     10    CONTINUE
835     C
836     C ----- IF END OF PLOT CALL FOR NEXT USER COMMAND
837     C ----- OR RETURN TO THE DAP FOR THE NEXT PLANE
838     C
839         IF(INSTR.EQ.0)CALL INSTRUCTION
840         RETURN
841         END
842     C
843     C
844     C
845     C
846     C
847         SUBROUTINE INSTRUCTION
848     C
849     C ----- THIS SUBROUTINE IS CALLED EACH TIME A USER
850     C ----- COMMAND IS REQUIRED. THE INFORMATION R'Q'D
851     C ----- IS THE LEVEL OF DATA REQUIRED AND THE
852     C ----- COORD VALUE (IN RASTER) OF THE LOWER LEFT
853     C ----- HAND CORNER.
854     C
855     C ----- INSTRUCTION 999 MEANS STOP THE PROGRAM
856     C
857         INTEGER*8 INT,IX,IY
858         COMMON/INST/INT,IX,IY
859         CALL END
860         WRITE(6,100)
861     100    FORMAT(' TYPE IN : INSTRUCTION IX IY ')
862         READ(5,*)INT,IX,IY
863         IF(INT.EQ.999)RETURN
864     C
865     C ----- PREPARE THE DISPLAY
866     C
867         CALL CLEAR
868         RETURN
869         END
```

END OF FILE

ANNEX 6d

DAP-FORTRAN source listing of software to isolate sub-parcel data from OS large scale map images in raster form.

See section 6.4.2 of text for details

```
1      C ***** T H I S       I S       P A R C E L 5 1 2
2      C ***** A N D         S U B P A R C E L 5 1 2
3      C
4          PROGRAM PARCEL512
5      C
6      C *** THIS IS THE HOST FORTRAN SECTION OF THE
7      C *** SUB-PARCEL EXTRACTION SOFTWARE. ITS PURPOSE
8      C *** IS TO CONTROL THE DAP MODULES IN SUBDIVIDING
9      C *** A COMPLETE OS RASTER IMAGE INTO A SERIES OF
10     C *** SUB-PARCEL UNITS. THE RASTER INPUT DATA ARE
11     C *** READ ON CHANN. 10 AND THE LEVEL 6 INDEX IS
12     C *** READ ON CHANN. 7
13     C *** SEE SECTION 6.4.2 OF TEXT FOR DESCRIPTION
14     C
15     C
16     C ----- DECLARATION OF INTEGER*8 MEANS THAT
17     C ----- NO CONVERSION TO DAP FORMAT IS NECESSARY
18     C
19         INTEGER*8 NFC,SUM6,NFIRST
20         INTEGER INDEX(4096)
21         COMMON/FC/NFC
22     C
23     C ----- FILE IS DECLARED SUITABLY LARGE AND
24     C ----- IS, IN EFFECT, THE DAP TO THIS PROGRAM
25     C
26         CHARACTER*64 FILE(14000)
27         COMMON/DATA/FILE
28         COMMON/IND/INDEX,SUM6,NFIRST
29         EXTERNAL VECTOUT
30     C
31     C ----- THE USE OF AN ERROR TRAP ENABLES A SNEAKY
32     C ----- REVERT BACK FROM THE DAP TO THE 2980
33     C
34     C ----- THE FOLLOWING THREE LINES ARE JUST
35     C ----- VME/B MANDATORY FOR ERROR TRAP CATCHING
36     C ----- SUB "VECTOUT" WILL RUN ON EACH ERROR
37     C
38         CALL ICL9HERRORTRAPDR(VECTOUT,IRES)
39         IF(IRES.NE.0)WRITE(6,1)
40         1 FORMAT(' SET VECTOUT UNSUCCESSFUL')
41     C
```



```
42 C ----- ASK THE USER TO REQUEST WHICH F/C HE WISHES
43 C
44 WRITE(6,100)
45 100 FORMAT(' WHICH FEATURE CODE?')
46 READ(5,*)NFC
47 C
48 C ----- INPUT OF THE INDEX LEVEL 6 FILE
49 C
50 NF=1417*8
51 READ(7,1000)(FILE(I),I=1,NF)
52 1000 FORMAT(1A64)
53 C
54 C ----- OPEN AND LINK UP CHANNEL 10 TO APPROPRIATE
55 C ----- NFC LEVEL 12 (DAP FORMAT) FILE
56 C
57 CALL CONNECTFILE(NFC)
58 C
59 C ----- EXAMINE DESIRED LEVEL 6 FILE, SHUNT
60 C ----- POSITIONAL INDEX TO TOP OF LONG VECTOR
61 C
62 CALL GETNFC
63 C
64 C
65 C ----- SUM6 IS THE NO. OF SET LEVEL 6 PIXELS
66 C
67 NF=SUM6*8
68 C
69 C ----- READ IN THE DESIRED LEVEL 12 DATA
70 C
71 READ(10,1000)(FILE(I),I=1,NF)
72 C
73 C ----- SET UP THE SIGMA GRAPHICS TERMINAL
74 C
75 CALL START
76 CALL SETBUFSIZE(1024,6)
77 C
78 C ----- PREPARE LEVEL 12 FILE TO A LEVEL 9 REP.
79 C
80 CALL SETUP512
81 C
82 C ----- START IDENTIFYING SUB-PARCELS ON DAP
83 C
84 CALL SUBPARCEL512
85 CALL END
86 STOP
87 END
88 C
89 C
90 C
91 C
92 C
93 ENTRY SUBROUTINE SETUP512
```

```
94      C
95      C ----- THIS IS A SUBROUTINE TO CONVERT LEVEL 12
96      C ----- REP. OF RASTER IMAGE TO LEVEL 9
97      C
98          INTEGER INDEX(,),SUM6
99          COMMON/IND/INDEX,SUM6
100     C
101     C ----- SEE ANNEX 6C FOR THIS SUB
102     C
103          CALL GRID512(INDEX,SUM6)
104          RETURN
105          END
106     C
107     C
108     C
109     C
110     C
111          ENTRY SUBROUTINE SUBPARCEL512
112     C
113     C ----- THIS IS THE ENTRY SUBROUTINE FOR THE
114     C ----- SUB-PARCEL EXTRACTION FOR THE LEVEL 9 DATA
115     C ----- IT HANDLES ANY INITIALISING AND SUBR CALLS
116     C
117          LOGICAL NSIDE(,)
118          LOGICAL CORNERS(,)
119          LOGICAL PADDING(,,22)
120          INTEGER BBUFF(,,11)
121          EQUIVALENCE (BBUFF,BUFF)
122          LOGICAL EDGE(,,4),START(,),BOUND(,),NBOUND(,)
123          LOGICAL SIDE(,),MAP(,,8,8)
124          LOGICAL POINT(,)
125          INTEGER*2 V(,),XC(,),YC(,)
126          INTEGER*2 P(256),L(256),PARCEL(62,2),NPP,NP,IB
127          INTEGER C(,),BUFF(,,10)
128          INTEGER*2 OLDP,NCOOORD
129          INTEGER*2 II,JJ
130     C
131     C ----- SET UP AN ARRAY 'CORNERS' SO THAT ONLY EACH
132     C ----- CORNER ELEMENT IS SET
133     C
134          DATA CORNERS/#800000000000000001,
135          1          62* #000000000000000000,
136          1          #800000000000000001/
137          COMMON/XY/XC,YC
138          COMMON/IMAGE/IX,IY
139          COMMON/XYBUFF/BUFF,P,L,PARCEL,NPP,NP,IB,
140          1PADDING
141          COMMON/MAPS9/MAP
142          EXTERNAL LOGICAL MATRIX FUNCTION NBH
143     C
144     C ----- SET-UP TO MATRICES XC AND YC TO AID
145     C ----- VECTORISING OF SUB-PARCELS. XC CONTAINS
```

```
146 C ----- SUCCESSIVE ROWS OF 1,2,3,....64 AND
147 C ----- YC COLS OF 1,2,3,...64
148     N=1
149     V=1
150     DO 100 I=1,6
151     V=V+SHRP(V,N)
152     100 N=N*2
153     V=V-1
154     XC=MATR(V)
155     YC=MATC(REV(V))
156 C
157 C ----- INITIALISE BUFF WHICH IS A COLLECTION OF
158 C ----- TO STORE THE VECTOR DATA BEFORE MATRICES
159 C ----- I/O TO A FILE ON THE 2980
160 C
161     NP=0
162     DO 200 I=1,11
163     200 BBUFF(,,I)=0
164     IB=1
165     NPP=1
166     P(1)=1
167     L(1)=0
168     OLDP=1
169 C
170     DO 300 I=1,8
171     DO 300 J=1,8
172 C
173 C -----TRANSFORM MAP SO THAT ITS APPEARANCE IS THE
174 C -----CORRECT WAY ROUND
175 C
176     300 MAP(,,I,J)=TRAN(MAP(,,I,J))
177 C
178 C ----- PUT A BORDER RIGHT AROUND THE EDGE OF MAP
179 C
180     CALL BORDERMAP512(MAP)
181 C
182 C ----- SET UP EDGE WHICH IS A COLLECTION OF 4
183 C ----- LOGICAL MATRICES SUCH THAT EACH WILL
184 C ----- HAVE 1 OF 4 POSSIBLE EDGES SET ONLY
185 C
186     CALL SETEDGE(EDGE)
187 C
188 C ----- DAP IMAGE BY DAP IMAGE, START TO LOOK FOR
189 C ----- SUB-PARCELS
190 C
191     DO 10 I=1,8
192     DO 10 J=1,8
193     DO 40 LL=1,100
194 C
195 C ----- BOUND IS THE TEMP LOGICAL MATRIX WHICH
196 C ----- STORES THE CURRENT DAP IMAGE UNDER SCRUTINY
197 C
```

```
198         BOUND=MAP(,,I,J)
199     C
200     C ----- START IS A LOGICAL MATRIX IN WHICH THE
201     C ----- POINT EXPANSION WILL TAKE PLACE DURING
202     C ----- THE PARCEL EXTRACTION PROCESS
203     C
204     C ----- SET THE FIRST POINT IN THE FIRST NEXT
205     C ----- AVAILABLE PARCEL TO TRUE IN READINESS FOR
206     C ----- POINT EXPANSION.
207     C
208         START=FRST(.NOT.BOUND)
209     C
210     C ----- IF THERE ARE NO AVAILABLE TRUES LEFT IN
211     C ----- START THEN THIS DAP IMAGE HAS BEEN
212     C ----- FINISHED SO GO TO THE NEXT
213     C
214         IF(ALL(.NOT.START))GOTO 10
215         NP=NP+1
216     C
217     C ----- START THE POINT EXPANSION PROCESS UP TO THE
218     C ----- PARCEL BDY. 130 IS AN EMPIRICAL NUMBER TO
219     C ----- ENSURE THAT ALL DAP IMAGE IS SEARCHED.
220     C
221         POINT=START
222         DO 20 K=1,130
223     20 START(.NOT.BOUND)=NBH(START)
224         CALL VECTOR(START,BOUND,I,J,POINT)
225     C
226     C ----- AMEND THIS VERSION OF MAP NOW BY
227     C ----- 'PAINTING-OUT' THE SUB-PARCEL WHICH
228     C ----- HAS CURRENTLY BEEN EXTRACTED.
229     C
230         MAP(,,I,J)=BOUND.OR.START
231     C
232     C ----- NOW LOOK TO SEE IF THE CURRENT SUB-PARCEL
233     C ----- OVERLAPS INTO ANY ADJACENT DAP IMAGE
234     C
235         DO 30 K=1,4
236         SIDE=START.AND.EDGE(,,K).AND..NOT.CORNERS
237     C
238     C ----- IF THERE IS NO OVERLAP GO BACK AND LOOK FOR
239     C ----- NEXT SUBPARCEL IN THIS DAP IMAGE
240     C
241         IF(.NOT.ANY(SIDE))GOTO 30
242     C
243     C ----- SEARCH SIDE BY SIDE FOR THE PARCEL IN THE
244     C ----- NEXT ADJACENT DAP IMAGE. II AND JJ ARE
245     C ----- NEEDED TO KEEP THE RECURSIVE STACK AS SMALL
246     C ----- AS POSSIBLE WHEN SUB NEXTPLANE512
247     C ----- IS CALLED RECURSIVELY
248     C
249         II=I
```

```
250          JJ=J
251          GOTO(1,2,3,4),K
252      1    II=I-1
253          SIDE=SHNC(SIDE)
254          GOTO 25
255      2    JJ=J+1
256          SIDE=SHEC(SIDE)
257          GOTO 25
258      3    II=I+1
259          SIDE=SHSC(SIDE)
260          GOTO 25
261      4    JJ=J-1
262          SIDE=SHWC(SIDE)
263      C
264      C
265      C ----- GO SEARCH NEXT DAP IMAGE FOR OVERLAP PARCEL
266      C
267          25 CALL NEXTPLANE512(MAP, II, JJ, KK, SIDE, EDGE,
268             INBOUND, CORNERS, NSIDE)
269          30 CONTINUE
270             PARCEL(NP, 1)=OLDP
271             OLDP=OLDP+PARCEL(NP, 2)
272             CALL DISPLAYPARCEL512
273          40 CONTINUE
274          10 CONTINUE
275             RETURN
276             END
277      C
278      C
279      C
280      C
281      C
282          SUBROUTINE BORDERMAP512(MAP)
283      C
284      C ----- THIS IS A SUBROUTINE WHICH TAKES A
285      C ----- COLLECTION OF 8 X 8 DAP IMAGES AND SETS
286      C ----- THE OUTER BORDER OF THE WHOLE SET
287      C ----- I.E. IT SETS AN 8 X 8 BORDER
288      C
289          LOGICAL MAP(,,8,8), EDGE(,,4)
290      C
291          CALL SETEDGE(EDGE)
292          DO 10 K=1,8
293             MAP(,,1,K)=MAP(,,1,K).OR.EDGE(,,1)
294             MAP(,,8,K)=MAP(,,8,K).OR.EDGE(,,3)
295             MAP(,,K,1)=MAP(,,K,1).OR.EDGE(,,4)
296      10    MAP(,,K,8)=MAP(,,K,8).OR.EDGE(,,2)
297             RETURN
298             END
299      C
300      C
301      C
```

```
302 C
303 C
304 SUBROUTINE SETEDGE(EDGE)
305 C
306 C ----- THIS IS A SUBROUTINE WHICH SETS UP A
307 C ----- COLLECTION OF 4 MATRICES AND SETS ONE EDGE
308 C ----- TO TRUE IN EACH OF THE 4 ENSURING THAT
309 C ----- THERE IS NO REPETITION OF THE SET SIDE
310 C
311 LOGICAL EDGE(,,4)
312 EDGE(,,1)=.FALSE.
313 EDGE(,,2)=.FALSE.
314 EDGE(,,3)=.FALSE.
315 EDGE(,,4)=.FALSE.
316 EDGE(1,,1)=.TRUE.
317 EDGE(64,,3)=.TRUE.
318 EDGE(,64,2)=.TRUE.
319 EDGE(,1,4)=.TRUE.
320 RETURN
321 END
322 C
323 C
324 C
325 C
326 C
327 LOGICAL MATRIX FUNCTION NBH(LM)
328 C
329 C ----- THIS IS A LOGICAL FUNCTION WHICH TAKES A
330 C ----- LOGICAL MATRIX AS INPUT AND SHIFTS THE
331 C ----- CONTENTS IN 4 DIRECTIONS THEREBY SETTING
332 C ----- THE 4 NEAREST NEIGHBOURS IN THE ORIGINAL
333 C ----- IMAGE. REPETITIVE USE MEANS THAT A SINGLE
334 C ----- ORIGINAL POINT WILL EXPAND TO EVENTUALLY
335 C ----- COVER THE ENTIRE MATRIX.
336 C
337 LOGICAL LM(,)
338 NBH=LM.OR.LM(+,).OR.LM(,+).OR.LM(-,).OR.LM(,-)
339 RETURN
340 END
341 C
342 C
343 C
344 C
345 C
346 SUBROUTINE VECTOR(START,BOUND,II,JJ)
347 C
348 C ----- THIS IS A SUBROUTINE TO TAKE ISOLATED
349 C ----- SUB-PARCEL DATA HELD IN DAP PLANE START
350 C ----- GENERATE ITS BOUNDARY PIXELS INTO PAR
351 C ----- AND CONVERT EACH PIXEL TO X,Y VALUES
352 C ----- IN INTEGER MATRIX C
353 C ----- THESE VALUES ARE BUFFERED TOGETHER FOR
```

```
354 C ----- STORAGE IN BUFF FOR ARCHIVE OR DISPLAY
355 C
356 LOGICAL LC(,,32)
357 LOGICAL START(,),BOUND(,),PAR(,)
358 INTEGER*2 XC(,),YC(,),P(256),L(256),
359 1PARCEL(62,2),X(,),Y(,),NPP,NP
360 INTEGER BUFF(,,10),C(,)
361 INTEGER*2 II,JJ,IB
362 EXTERNAL INTEGER MATRIX FUNCTION IPACK
363 EXTERNAL LOGICAL MATRIX FUNCTION NBH
364 COMMON/XY/XC,YC
365 COMMON/XYBUFF/BUFF,P,L,PARCEL,NPP,NP,IB
366 EQUIVALENCE (LC,C,X),(LC(,,17),Y)
367 C
368 C ----- GENERATE X AND Y MATRICES WITH
369 C ----- ELEMENT COORDINATES IN THEM
370 C
371 Y=(JJ-1)*64+XC
372 X=(8-II)*64+YC
373 C
374 C ----- INITIALISE PAR - A MATRIX TO HOLD
375 C ----- THE SUB-PARCEL BOUNDARY
376 C
377 PAR=.FALSE.
378 C
379 C ----- GENERATE THE SUB-PARCEL BDY. INTO PAR
380 C
381 PAR(BOUND)=NBH(START)
382 NCOORD=SUM(PAR)
383 C
384 C ----- SET-UP POINTERS FOR THE START AND END
385 C ----- OF VECTOR CHAINS IN THE SEQUENTIAL
386 C ----- BUFFER STORE
387 C
388 IP=P(NPP)+L(NPP)
389 IS=IP-(IP-1)/4096*4096
390 NPP=NPP+1
391 P(NPP)=IP
392 L(NPP)=NCOORD
393 PARCEL(NP,2)=PARCEL(NP,2)+NCOORD
394 C
395 C ----- CONVERT PIXEL BOUNDARY TO COORDINATES
396 C
397 C(.NOT.PAR)=0
398 C
399 C ----- SHUNT THESE COORDS TO TOP OF C
400 C
401 C=SHRC(IPACK(C),IS-1)
402 C
403 C ----- STORE THEM IN THE SEQUENTIAL BUFFER
404 C
405 BUFF(.NOT.ELSL(1,IS-1),IB)=C
```

```
406      IF((IS+NCOORD-1).LE.4096)RETURN
407      IB=IB+1
408      BUFF(,,IB)=C
409      RETURN
410      END
411      C
412      C
413      C
414      C
415      C
416      SUBROUTINE NEXTPLANE512(MAP,II,JJ,KK,START,
417      1EDGE,BOUND,CORNERS,SIDE)
418      C
419      C ----- THIS IS A SUBROUTINE WHICH CAN BE CALLED
420      C ----- RECURSIVELY AND IS DESIGNED TO SEARCH
421      C ----- ADJACENT DAP PLANES WHEN SUBPARCEL OVERLAPS
422      C ----- INTO THEM. IT WORKS EXACTLY THE SAME AS
423      C ----- THE ENTRY SUB SUBPARCEL512 WHICH CONTROLS
424      C ----- THE STARTING POINT OF THE RECURSION.
425      C
426      C
427      C
428      LOGICAL SIDE(,)
429      LOGICAL CORNERS(,)
430      INTEGER*2 III,JJJ
431      INTEGER*2 II,JJ
432      LOGICAL MAP(,,8,8),START(,),BOUND(,),EDGE(,,4)
433      EXTERNAL LOGICAL MATRIX FUNCTION NBH
434      BOUND=MAP(,,II,JJ)
435      IF(.NOT.ANY(.NOT.CORNERS.AND.START.AND.
436      1.NOT.BOUND))RETURN
437      DO 20 KK=1,130
438      20 START(.NOT.BOUND)=NBH(START)
439      CALL VECTOR(START,BOUND,II,JJ)
440      MAP(,,II,JJ)=BOUND.OR.START
441      DO 30 KK=1,4
442      SIDE=START.AND.EDGE(,,KK).AND..NOT.CORNERS
443      IF(.NOT.ANY(SIDE))GOTO 30
444      III=II
445      JJJ=JJ
446      GOTO (1,2,3,4),KK
447      1 III=III-1
448      SIDE=SHNC(SIDE)
449      GOTO 35
450      2 JJJ=JJJ+1
451      SIDE=SHCC(SIDE)
452      GOTO 35
453      3 III=III+1
454      SIDE=SHSC(SIDE)
455      GOTO 35
456      4 JJJ=JJJ-1
457      SIDE=SHWC(SIDE)
```



```
458      35 CALL NEXTPLANE512(MAP, III, JJJ, KK, SIDE, EDGE,
459      1BOUND, CORNERS, START)
460      30 CONTINUE
461      RETURN
462      END
463      C
464      C
465      C
466      C
467      C
468      SUBROUTINE DISPLAYPARCEL512
469      C
470      C ----- THIS IS A SUBROUTINE TO UNCOMPACT THE
471      C ----- VECTOR BUFFER BUFF AND REORGANISE THE
472      C ----- CONTENTS INTO 'OUT' IN A FORM SUITED
473      C ----- FOR DISPLAY ON A SIGMA RASTER DISPLAY
474      C
475      INTEGER*2 P(256), L(256), PARCEL(62, 2), NPP, NP, IB
476      INTEGER BUFF(, , 10), OUT(, ), NC
477      COMMON/XYBUFF/BUFF, P, L, PARCEL, NPP, NP
478      COMMON/XYOUT/OUT, NC, ID1, ID2, NNP
479      C
480      C ----- ESTABLISH THE BUFFER POINTERS
481      C
482      NNP=NP
483      IP=PARCEL(NP, 1)
484      IS=IP-(IP-1)/4096*4096
485      IB=(IP-1)/4096+1
486      NC=PARCEL(NP, 2)
487      C
488      C ----- UNPACK THE BUFFER FOR THIS DAP PLANE
489      C
490      OUT=SHLP(BUFF(, , IB), IS-1)
491      IF (IS+NC-1.LE.4096) GOTO 10
492      OUT(.NOT.ELSL(1, 4096-IS+1).AND..NOT.ELSL
493      1(NC+1, 4096))=SHRP(BUFF(, , IB+1), 4096-IS+1)
494      C
495      C ----- CONVERT FROM DAP FORMAT TO 2980 FORMAT
496      C
497      10 CALL CONVMF4(OUT)
498      C
499      C ----- CALL FOR I/O FROM 2980 FOR SIGMA
500      C
501      ERROR 8999
502      RETURN
503      END
504      C
505      C
506      C
507      C
508      C
509      SUBROUTINE VECTOUT(L, NUM)
```

```
510      C
511      C ----- THIS IS 2980 FORTRAN TO DUMP THE SCAN-LINE
512      C ----- ORDERED SUB-PARCEL BOUNDARY COORDINATES ON
513      C ----- THE SIGMA SCREEN
514      C
515          INTEGER*8 NPT
516          CHARACTER*1 L
517          INTEGER*2 OUT(2,4096)
518          INTEGER NUM
519          COMMON/XYOUT/OUT,NPT
520      C
521      C ----- VME/B MANDATORY FOR ERROR TRAP CATCHING
522      C
523          IF (L.NE.'S') RETURN
524          IF (NUM.NE.8999) RETURN
525          CALL ICL9HERRORTABLE(0,-1,-1,0,0,0,0,-1,0,
526          1-1,1,IRES)
527          IF (NPT.GT.4096)NPT=4096
528      C
529      C ----- NPT IS NUMBER OF COORDS IN BOUNDARY
530      C
531          DO 10 I=1,NPT
532      C
533      C ----- GENERATE SIGMA COORD SYSTEM FOR
534      C ----- DAP PLANE 'OUT'
535      C
536          IX=511-OUT(1,I)
537          IY=OUT(2,I)
538      C
539      C ----- MOVE TO POSITION X,Y (PEN UP)
540      C
541          CALL MOVE(IY,IX,.FALSE.)
542      C
543      C ----- DRAW A DOT WITH COLOUR OPTION 8
544      C
545          CALL OUTLINE('8',1)
546      10  CONTINUE
547      C
548      C ----- BACK TO DAP FOR NEXT DAP PLANE
549      C
550          RETURN
551          END
END OF FILE
```

ANNEX 6e

DAP-FORTRAN source listing of software to perform a raster-to-vector conversion on the sub-parcel data for OS large scales data in DAP format.

See section 6.4.2 of text for full description

```
1          SUBROUTINE GINOV
2          C
3          C *** THIS IS A SUBROUTINE TO GENERATE VECTOR CHAINS
4          C *** SUITABLE FOR INPUT TO A PLOTTING PACKAGE SUCH
5          C *** AS GINO OR GHOST ETC..
6          C
7          C *** IT TAKES AS INPUT A SUB-PARCEL BOUNDARY IN
8          C *** PIXEL FORM OVERLAPPING UP TO 8 X 8 DAP
9          C *** PLANES WHICH ARE HANDLED ONE AT A TIME IN
10         C *** THE LOGICAL MATRIX "POLY". THE PARCEL
11         C *** ENTERS THE SUBROUTINE THROUGH A LOGICAL
12         C *** MATRIX FUNCTION "WINDOW" WHICH GIVES THE
13         C *** APPROPRIATE DAP IMAGE IN POLY. THE VECTOR
14         C *** IS FOLLOWED AND AS IT APPROACHES THE EDGE
15         C *** OF THE DAP PLANE THE ADJACENT HALF DAP IMAGE
16         C *** IS BROUGHT INTO POLY. THUS, INSTEAD OF USING
17         C *** I,J COORDS (1-8) TO REPRESENT THE CURRENT
18         C *** WINDOW POSITION, IW,JW ARE USED (1-17) WHICH
19         C *** ARE UNITS OF HALF DAP IMAGES.
20         C
21         INTEGER*2 XLINE(4096)
22         LOGICAL SUB(,,8,8)
23         LOGICAL POLY(,),START(,),ALLPTS(,),EDGE(,,4)
24         INTEGER*4 C(,),LINE(4096),POINTER(64)
25         EXTERNAL LOGICAL MATRIX FUNCTION NBH,WINDOW,
26         1SHIFT
27         EXTERNAL INTEGER*4 MATRIX FUNCTION
28         1WINDOWCOORDS
29         COMMON/SUBPARCEL/SUB
30         COMMON/VECTORS/LINE,POINTER
31         EQUIVALENCE (XLINE,LINE)
32         C
33         C ----- INITIALISE LINE BUFFER POINTER AND COUNTER
34         C
35         LP=1
36         IL=1
37         IPTR=1
38         C
39         C ----- SEE ANNEX 6D FOR THIS SUBROUTINE
40         C
```

```
41          CALL SETEDGE(EDGE)
42          C
43          C ----- START THE SEARCH CYCLE FOR PARCEL IN FIRST
44          C ----- DAP IMAGE AND CYCLE UNTIL FOUND
45          C
46          DO 10 I=1,8
47          DO 10 J=1,8
48          C
49          C ----- CONVERT DAP IMAGE COORDS TO WINDOW COORDS
50          C
51          IW=2*(I-1)+2
52          JW=2*(J-1)+2
53          C
54          C ----- BRING APPROPRIATE WINDOW INTO POLY AND PUT
55          C ----- ABSOLUTE COORDS FOR THAT WINDOW INTO C
56          C
57          POLY=WINDOW(IW,JW)
58          C=WINDOWCOORDS(IW,JW)
59          LP100=LP+100
60          DO 20 LL=LP,LP100
61          C
62          C ----- SET START TO HAVE ONE PIXEL SET ON POLY. BDY
63          C
64          START=FRST(POLY)
65          ALLPTS=START
66          TRACE 2(POLY,ALLPTS,IW,JW)
67          C
68          C ----- IF THIS DAP PLANE IS EMPTY SKIP TO NEXT
69          C
70          IF(ALL(.NOT.START))GOTO 10
71          C
72          C ----- PAINT OUT IN POLY THE CURRENT PIXEL SET IN
73          C ----- START
74          C
75          POLY=POLY.AND..NOT.START
76          C
77          C-----TEST FOR 1 PIXEL POLYGON CASE
78          C
79          IF (.NOT.ANY(POLY))GOTO 10
80          C
81          C ----- PUT THE COORDS OF CURRENT PT IN START INTO
82          C ----- STORAGE BUFFER 'LINE' AND INCREMENT POINTER
83          C
84          LINE(IL)=C(START)
85          IL=IL+1
86          C
87          C ----- ARBITRARILY LARGE NO. TO COUNT SUB-PARCELS
88          C
89          DO 30 LLL=1,14096
90          C
91          C ----- ARE WE AT THE EDGE OF A DAP PLANE?
92          C ----- IGNORE THIS LOOP IF WE AREN'T
```

```
93      C
94      50  DO 60 IDIR=1,8
95          IF (ANY(POLY.AND.SHIFT(START, IDIR))) GOTO 70
96      60  CONTINUE
97      C
98      C ----- ARE WE AT EDGE OF WINDOW ? IF NOT GOTO 40
99      C
100     DO 40 K=1,4
101     IF(.NOT.ANY(START.AND.EDGE(,,K)))GOTO 40
102     C
103     C ----- PAINT OUT THE NEAREST NEIGHBOURS OF LINE
104     C ----- DETECTED THUS FAR TO REMOVE PROBLEM OF
105     C ----- DOUBLE THICKNESS PIXEL-WIDTHS
106     C ----- SEE ANNEX 6D FOR NBH DESCRIPTION
107     C
108     POLY(NBH(ALLPTS).AND..NOT.ALLPTS)=.FALSE.
109     C
110     C ----- PUT UPDATED VERSION OF POLY AFTER PAINT OUT
111     C ----- BACK INTO STORAGE
112     C
113     CALL STOREWINDOW(POLY, IW, JW)
114     C
115     C ----- MOVE CURRENT PT (START) BY HALF DAP IMAGE
116     C ----- IN APPROPRIATE DIRECTION FOR NEXT WINDOW
117     C
118     GOTO (1,2,3,4),K
119     1  IW=IW-1
120     START=SHSP(START,32)
121     GOTO 5
122     2  JW=JW+1
123     START=SHWP(START,32)
124     GOTO 5
125     3  IW=IW+1
126     START=SHNP(START,32)
127     GOTO 5
128     4  JW=JW-1
129     START=SHEP(START,32)
130     C
131     C ----- BRING IN NEW WINDOW AND UPDATE COORDS IN C
132     C
133     5  POLY=WINDOW(IW, JW)
134     C=WINDOWCOORDS(IW, JW)
135     ALLPTS=START
136     TRACE 2(POLY, ALLPTS, IW, JW)
137     GOTO 50
138     40 CONTINUE
139     POINTER(IPTR)=IL
140     IPTR=IPTR+1
141     TRACE 1(POLY, START, IW, JW, ALLPTS)
142     GOTO 20
143     C
144     C ----- SEARCH ALL THE 8 NEIGHBOURS FOR THE NEXT
```

```
145 C ----- PT IN THE CHAIN. CLOCKWISE SEARCH 1=NORTH
146 C
147 C
148 C ----- PAINT OUT CURRENT START PT FROM POLY
149 C
150 C     70 POLY=POLY.AND..NOT.START
151 C
152 C ----- MOVE CURRENT START PT INTO THE APPROPRIATE
153 C ----- NEW PIXEL POSITION IN DIRECTION IDIR
154 C
155 C     START=SHIFT(START, IDIR)
156 C
157 C ----- UPDATE ALLPTS
158 C
159 C     ALLPTS=ALLPTS.OR.START
160 C     TRACE 2(POLY, START, ALLPTS, IW, JW)
161 C
162 C -----PUT ASSOC. ABSOLUTE COORDS IN STORAGE BUFFER
163 C ----- LINE AND UPDATE COUNTER
164 C
165 C     LINE(IL)=C(START)
166 C     IL=IL+1
167 C     30 CONTINUE
168 C     POINTER(IPTR)=IL
169 C     IPTR=IPTR+1
170 C     20 CONTINUE
171 C     TRACE 1(IPTR, IL)
172 C     10 CONTINUE
173 C     POINTER(IPTR)=-1
174 C
175 C ----- INVOKE AN ERROR CONDITION TO FORCE 2980 TO
176 C ----- TAKE OVER FOR I/O CONTROL
177 C
178 C     ERROR 8999
179 C     RETURN
180 C     END
181 C
182 C
183 C
184 C
185 C
186 C     LOGICAL MATRIX FUNCTION WINDOW(IW, JW)
187 C
188 C ----- THIS FUNCTION RETURNS A WINDOW FROM AN 8x8
189 C ----- COLLECTION OF DAP IMAGES WITH COORDS IW, JW,
190 C ----- WHICH CAN BE INCREMENTED BY HALF A DAP
191 C ----- PLANE AT A TIME.
192 C ----- AT WORST, THE PARTITIONING COULD INVOLVE
193 C ----- TAKING QUADRANTS FROM THE ADJACENT PLANES
194 C ----- TO PUT INTO WINDOW
195 C
196 C     LOGICAL SUB(, , 8, 8), IL, JL, A(, ), B(, )
```

```
197          COMMON/SUBPARCEL/SUB
198          EXTERNAL LOGICAL MATRIX FUNCTION W
199      C
200      C ----- ASCERTAIN WHETHER IW,JW ARE ODD OR EVEN
201      C ----- ODD SUGGESTS PARTITIONING WILL BE NECESSARY
202      C
203          IL=(IW-IW/2*2.EQ.0)
204          JL=(JW-JW/2*2.EQ.0)
205          A=.FALSE.
206          B=.FALSE.
207      C
208      C ----- IF IW, JW ARE ODD GOTO 10 OTHERWISE BRING
209      C ----- BACK A COMPLETE DAP IMAGE
210      C
211          IF(.NOT.(IL.AND.JL))GOTO 10
212          WINDOW=SUB(,,IW/2,JW/2)
213          RETURN
214      C
215      C ----- DAP PARTITIONING IS REQUIRED. HERE ARE
216      C ----- THE THREE CASES
217      C
218          10 IF(.NOT.(IL.AND..NOT.JL))GOTO 20
219      C
220      C ----- VERTICAL PARTITION INTO HALVES
221      C
222          IF(JW.NE.1)A=SHWP(SUB(,,IW/2,JW/2),32)
223          IF(JW.NE.17)B=SHEP(SUB(,,IW/2,JW/2+1),32)
224          WINDOW=A.OR.B
225          RETURN
226          20 IF(.NOT.(.NOT.IL.AND.JL))GOTO 30
227      C
228      C ----- HORIZONTAL PARTITION INTO HALVES
229      C
230          IF(IW.NE.1)A=SHNP(SUB(,,IW/2,JW/2),32)
231          IF(IW.NE.17)B=SHSP(SUB(,,IW/2+1,JW/2),32)
232          WINDOW=A.OR.B
233          RETURN
234      C
235      C ----- RECURSIVE CALL OF THIS FUNCTION TO HANDLE
236      C ----- THE 4 QUADRANT CASE WITH HORIZONTAL
237      C ----- AND VERTICAL PARTITION
238      C
239          30 IWm1=IW-1
240          IWp1=IW+1
241          IF(IW.NE.1)A=W(IWm1,JW)
242          IF(IW.NE.17)B=W(IWp1,JW)
243          WINDOW=SHNP(A,32).OR.SHSP(B,32)
244          RETURN
245          END
246      C
247      C
248      C
```

```
249 C
250 C
251     INTEGER*4 MATRIX FUNCTION WINDOWCOORDS(IW,JW)
252 C
253 C ----- THIS IS A FUNCTION TO RETURN THE ABSOLUTE
254 C ----- COORDS OF ALL PIXELS IN A PARTICULAR WINDOW
255 C ----- WITH REFERENCE TO THE 8 X 8 ORIGINAL FRAME
256 C
257     LOGICAL LC(,,32)
258     INTEGER*4 C(,)
259     INTEGER*2 XC(,),YC(,),X(,),Y(,)
260 C
261 C ----- XC AND YC ARE MATS. WITH X,Y COORDS IN THEM
262 C
263     COMMON/XY/XC,YC
264     EQUIVALENCE(LC,X,C),(LC(,,17),Y)
265 C
266 C ----- ADD ON THE EXCESS COORD VALUES FROM
267 C ----- THE SW CORNER OF THIS DAP PLANE
268 C
269     Y=(JW-2)*32+XC
270     X=(16-IW)*32+YC
271     WINDOWCOORDS=C
272     RETURN
273     END
274 C
275 C
276 C
277 C
278 C
279     SUBROUTINE STOREWINDOW(POLY,IW,JW)
280 C
281 C ----- THIS IS A SUBROUTINE TO RESTORE A DAP
282 C ----- WINDOW (POLY) INTO ITS APPROPRIATE
283 C ----- POSITION IN SUB. IT MAY REQUIRE SPLITTING
284 C ----- INTO CONSTITUENT HALVES (OR QUADS) AND
285 C ----- PUTTING INTO APPROPRIATE DAP PLANES
286 C ----- SINCE THE WINDOW MOVES IN UNITS OF HALF
287 C ----- A DAP PLANE
288 C
289     LOGICAL POLY(,),SUB(,,8,8),IL,JL
290     LOGICAL A(,),B(,)
291     EXTERNAL LOGICAL MATRIX FUNCTION WINDOW
292     COMMON/SUBPARCEL/SUB
293 C
294 C ----- ASCERTAIN WHETHER IW,JW ARE ODD OR EVEN
295 C ----- ODD SUGGESTS PARTITIONING WILL BE NECESSARY
296 C
297     IL=IW-IW/2*2.EQ.0
298     JL=JW-JW/2*2.EQ.0
299     IF(.NOT.(IL.AND.JL))GOTO 10
300 C
```



```
301 C ----- NO PARTITION NECESSARY - POLY FITS
302 C ----- EXACTLY INTO A "SUB" POSITION
303 C
304     SUB(, , IW/2, JW/2)=POLY
305     RETURN
306     10 IF(.NOT.(IL.AND..NOT.JL))GOTO 20
307 C
308 C ----- PARTITION NECESSARY - POLY REQUIRES
309 C ----- SPLITTING UP INTO HALVES WITH A
310 C ----- VERTICAL PARTITION LINE
311 C
312     A=SHEP(POLY, 32)
313     B=SHWP(POLY, 32)
314     IF(JW.NE.1)SUB(ALTC(32), IW/2, JW/2)=A
315     IF(JW.NE.17)SUB(.NOT.ALTC(32), IW/2, JW/2+1)=B
316     RETURN
317     20 IF(.NOT.(.NOT.IL.AND.JL))GOTO 30
318 C
319 C ----- PARTITION NECESSARY - POLY REQUIRES
320 C ----- SPLITTING UP INTO HALVES WITH A
321 C ----- HORIZONTAL PARTITION LINE
322 C
323     A=SHSP(POLY, 32)
324     B=SHNP(POLY, 32)
325     IF(IW.NE.1)SUB(ALTR(32), IW/2, JW/2)=A
326     IF(IW.NE.17)SUB(.NOT.ALTR(32), IW/2+1, JW/2)=B
327     RETURN
328 C
329 C ----- QUADRANT PARTITION CASE - THE CONTENTS
330 C ----- OF POLY ARE 4 QUADRANTS OF A DAP PLANE
331 C ----- TWO PARTITIONS ARE REQUIRED - VERTICAL
332 C ----- AND HORIZONTAL
333 C
334     30 IWPl=IW+1
335     IWm1=IW-1
336     IF (IWm1.EQ.0) GOTO 40
337     A=WINDOW(IWm1, JW)
338     A(ALTR(32))=SHSP(POLY, 32)
339     CALL STOREWINDOW(A, IWm1, JW)
340     40 IF (IWPl.EQ.18) RETURN
341     B=WINDOW(IWPl, JW)
342     B(.NOT.ALTR(32))=SHNP(POLY, 32)
343     CALL STOREWINDOW(B, IWPl, JW)
344     RETURN
345     END
346 C
347 C
348 C
349 C
350 C
351     SUBROUTINE GINOLIST(L, NUM)
352 C
```

```
353 C ----- THIS IS GINOLIST WRITTEN IN HOST FORTRAN
354 C ----- IT IS INVOKED EACH TIME ERROR 8999 OCCURS
355 C ----- IN GINOV. IT TAKES THE BUFFERED VECTOR
356 C ----- CHAINS FROM THE DAP IN "LINE" WITH ASSOC.
357 C ----- POINTERS TO START AND END OF EACH VECTOR
358 C ----- AND OUTPUT TO CHANN 6 X,Y COORDS IN FORMAT
359 C ----- 2I10. A -1 (INEG) IS PLACED BETWEEN EACH
360 C ----- VECTOR FEATURE
361 C
```

```
362 CHARACTER*1 L
363 INTEGER*2 LINE(4,4096)
364 INTEGER*8 POINTER(64)
365 INTEGER NUM
366 COMMON/VECTORS/LINE,POINTER
367 IF(L.NE.'S')RETURN
368 IF(NUM.NE.8999)RETURN
369 CALL ICL9HERRORTABLE(0,-1,-1,0,0,0,0,
370 1-1,0,-1,1,IRES)
371 INEG=-1
372 IEND=0
373 KOUNT=0
374 1 KOUNT=KOUNT+1
375 IPTR=POINTER(KOUNT)
376 IF(IPTR.LT.0)GOTO 999
377 ISTART=IEND+1
378 IEND=IPTR-1
379 WRITE(6,10)INEG
380 10 FORMAT(I10)
381 DO 20 I=ISTART,IEND
382 IX=LINE(3,I)
383 IY=LINE(4,I)
384 WRITE(6,15)IX,IY
385 15 FORMAT(2I10)
386 20 CONTINUE
387 GOTO 1
388 999 RETURN
389 END
```

END OF FILE