SPATIAL ANALYSIS TECHNIQUES

APPLIED TO

LOCAL CRIME PATTERNS

C. F. M. Brunsdon, B.Sc. (Hons), M.Sc.

Thesis submitted for the degree of PhD in the

Department Of Geography,

University Of Newcastle upon Tyne

September 1989

ACKNOWLEDGEMENTS

Thanks are owed to a great many people for their help and advice in the preparation of this Ph.D. thesis. To the Economic and Social Research Council and Northumbria Police for their generosity, through the award of a studentship enabling me to embark upon this research; To Dr. Stan Openshaw for his continued support and enthusiasm, and his willingness to discuss and constructively criticise all aspects of this study; To Inspector Jim Lillie for his support and advice when organising surveys and data collection from the Northumbria Police - without his contribution most of the reseach in this thesis would not have been possible; thanks are also due to many members of Newcastle University, particularly: John Goddard, Alf Thwaites, Martin Charlton, Julian Besag, Mike Coombes, Allan Gillard and all of the sectretarial staff at the Centre for Urban and Regional Development Studies. Finally I should like to thank all of those police officers who were willing to put aside time to complete questionaires for the map visualisation study in chapter 7 or take part in the user evaluation of the working system.

CONTENTS

A B S T R A C T

Despite growing crime rates, and increased computerisation of crime data within the police force, surprisingly little attention has been payed to techniques of analysis that could be applied to this data. This thesis investigates spatial analytical and statistical techniques which may be used for this task, and proposes a Bayesian forecasting technique, allowing the objective pattern detection mechanisms supplied by the quantitative examination of past data to be combined with the subjective knowledge of police officers. This method, along with others, is incorporated into a software package which may be run at a police station (Subdivision). Finally the software package is evaluated by members of the police force.

C H A P T E R 1

INTRODUCTION

1.1 Perspectives On Crime Pattern Analysis

The increasing availability of crime incidence data as a result of various database building projects raises important questions as to how such data might best be used for practical policing purposes. Traditionally, only a small amount of capital (in terms of money and resources) outlayed for the collation of these crime statistics has been allocated to management information tasks. All over Britain, police resource managers from subdivisional to force-wide levels have had to function without fully exploiting their comprehensive data resources.

The term Crime Pattern Analysis (CPA) has been used in policing for many years. Originally, because of the low number of incidents it was possible for beat policemen or detectives to identify patterns evolving in crimes reported based on personal memory, and to implement action accordingly. However, the large increase in the incidence of reported crimes in the 1970's and 1980s rendered informal Crime Pattern Analysis of this kind infeasible. A common response was to dedicate one police officer to identify patterns and draw them to the attention of police managers. However, eventually even this approach had serious shortcomings. The volume of data to be analysed is now too great for one officer within each subdivision to manage. A typical subdivision in

the Northumbria Police Force will have an annual crime load of 8,000 - 10,000 crimes per year.

The task of the analyst is to identify patterns in these crimes, perhaps identifying groups of related crimes of only a small number (perhaps as low as two or three), and there is no way of determining how many of these crimes are part of some pattern, and how many are merely "one-off" and of no predictive consequence.

It is obvious, therefore, that an individual officer carrying out Crime Pattern Analysis will require some tool to assist in handling the data. Thus, recently interest has been shown in computer-based systems, particularly using software for database management, and also in the potential utility of Geographical Information Systems (GIS). The latter are becoming widely adopted to manage map-related data, but like many of the tools currently available, they only offer limited analytical power. Mainly they present and manage the data, and the operators (in this case Crime Pattern Analysts) are left to identify the patterns themselves. A Police Foundation experiment involving an SIA developed GIS system (DataMap) based around an ordnance survey digital cartographic database for South Tyneside was found not to be particularly useful for crime pattern analysis. Although capable of flexible geographic data manipulation and display was possible, the analysis of this information was still entirely left to manual pattern detecting techniques.

However, computers are capable of far more than the input and selective output of raw data. Given the increasing richness of crime related data bases, with geographical referencing, it seems reasonable that more sophisticated tools be developed, to cope with the present and future needs of both the Police resource managers and Crime Pattern Analysts.

The objectives for the design of such techniques can be stated in terms of the development of computer models capable of predicting the incidence of crime in as much temporal and spatial details as is considered suitable. Ideally, such a system should run in real time, making use of crime report data as soon as it is collated, and be able to differentiate between essentially random events and those which appear to exhibit some regularity on either a time or space basis.

In order to achieve the above, techniques will need to be employed which analyse past crime data to identify such patterns. In addition to this, however, certain phenomena may influence crime rates on a short-term scale which may not be adequately captured in the data set. In making crime predictions, analysts will not rely solely on the data, but make use of other, more subjective knowledge in combination with this. For example, if it is known that an offender, say a burglar, who has been active in a particular area has recently been arrested, then it seems reasonable to assume that burglary rates in that area will drop in the near future. It would not be evident only from the inspection of past crime rates in the region (which would be high due to the burglars activity) that this sudden drop was about to happen. Thus, in addition

to the pattern analysis of the type described previously, some channels of input from the human analyst must still be left open.

It is noted that the tasks set out above are not easy, and to be executed successfully will require use of highly sophisticated methods with very specialised levels of expertise, and there is great conflict between the amount of work required to implement these methods and the availability of police research and development resources. Without work of this type, however, the alternative is to rely on simple methods that often fail to work well, and are justified only on the grounds of their simplicity. It is as a result of this conflict that this collaborative award between the ESRC and Northumbria Police has been granted.

There are theoretical and practical contributions that may be made from a wide range of fields, that can be applied to the problems stated here. These fields include probability theory, spatial analysis, management science and computer science. However, the problem may not be addressed by simply importing techniques developed elsewhere to the police data environment. Many of the standard techniques have never been applied in combination before, and there will be new techniques to be developed in addition to these. Also, it is to be remembered that the methods must be adapted to be suitable for use within a Police environment (rather than within a university or an operations research department of a large company).

The latter point is particularly important when complex statistical and probabilistic models are proposed for general police use. The "user

interface" has to be specially designed, made robust, and carefully developed to have the following characteristics; ease of use, and the provision of output that can be understood and communicated in police terms. Neither of these requirements preclude the use of complex methods, but do impose high standards on the "packaging" of these methods. The system must be designed to supply the required crime pattern analysis information and forecasts in terms relevant to the police user. From the users view the conceptually simple task of crime forecasting is being performed; although internally, sophisticated pattern analysis may be used to actually obtain the forecasts.

It is also important that the crime pattern questions that the system can answer are thought to be relevant, and so thought must be given to establishing what tasks are performed by the analyst. It is essential, therefore, to work in collaboration with the police, and to develop the system with some experience of the police environment. While the system requirements and interface design come from this environment, the knowledge to implement it is clearly interdisciplinary. It is hoped that over the study period the two components evolve together.

This introductory chapter will now go on to summarise the various aspects of this study, firstly by identifying the task of the crime pattern analyst, and then by considering particular issues in the implementation of the computerised system.

1.2 What Does a Crime Pattern Analyst Do?

In this section, a typical crime pattern analysis example will be given. The intention is to identify the skills that are used, and discover the strong and weak points. After this, models can be developed that may either replace those skills better suited to an automated approach, or provide backup to skills that human analysts are better at providing. A better understanding of what is involved can be gained by examining a hypothetical example of a crime pattern analysis task.

A list of crime incidents are given in figure 1.1. At the time of recording of the first incident, no pattern is discernible. Over a period of time more incidents accumulate incidents there is a possible link between three of them (A C D). The common features are that all of these crimes occurred overnight on terraced dwellings, by forcing a rear window. The analyst may now suspect a pattern. After three more offences, a similar pattern evolves in beat T5. There is also a pattern of days, in that all of the offences occur in the early hours of mid-week days. A further incident is on a semi-detached property, but the analyst is aware that this property has a rear yard, in common with the other properties. This suits the method of entry well. Also, in this pattern, if items of little or no value are stolen, another offence occurs quite soon after. In the case of more valuable items taken, a longer period between offences occurs.

It is also noted that offences on the east side of the major road in the locality involve theft of small items, whilst those on the west side involve

Figure 1.1
Crime Incident Reports

Crime Code	A	B	C	D	E	F
Beat	V3	V3	V3	V3	V3	V3
Date	4	5	5	6	11	10
Day	Tue	Wed	Wed	Thu	Tue	Mon
Times	7pm	4am-6pm	10pm-6am	12am-3am	--	1am
Dwelling Type	Terrace	Semi	Terrace	Terrace	Terrace	Ter.
Place Of Entry	Rear	Rear	Rear	Rear	Rear	Rear
Point Of Entry	Window	Door	Window	Window	Door	Window
Means Of Entry	Force	Force	Insecure	Force	Drill	Drill
Stolen (pounds)	5	420	0	0	400	60
Comments				Disturbed		

Crime Code	G	H	I
Beat	V3	T5	T5
Date	12	13	14
Day	Wed	Thu	Fri
Times	5pm-7am	11pm-8am	4pm-7am
Dwelling Type	Terrace	Terrace	Terrace
Place Of Entry	Rear	Rear	Rear
Point Of Entry	Window	Door	Window
Means Of Entry	Force	Force	Insecure
Stolen (pounds)	50	0	300
Comments			Video Taken

large items not simply concealed. The offender may therefore be based near here. The *criminal intelligence officer* may possibly be able to identify a suspect from this, but certainly a resource manager could be warned to deploy resources for the next mid-week period in the areas concerned.

This brief example highlights the type of skills applied by the crime pattern analyst. Some of these pertain to modelling the time and space constraints of offenders, while others involve detecting patterns in modus operandi and items stolen. In addition to this, a certain amount of intuitive input is required.

1.3 A "Smart" Computer Crime Incidence Forecaster

It is clear from the above example that simple GIS-based techniques largely fail to provide the lateral linkages that form a powerful part of the analysts work. There are two ways that this could be improved upon; (1) building a rules-based expert system (see, for example, Luger and Stubblefield, 1989); (2) developing a statistical system designed to work in parallel with the analyst. Option (1) lies some distance in the future, and suffers from the problem that to some extent crime risk is a random process, so imposing a deterministic model will yield unrealistically definite predictions (ie. no variability of outcome could be modelled).

This leaves option (2), using statistical forecasting, and pattern spotting techniques. In addition to this, however, there is the subjective knowledge factor described earlier. It seems essential to incorporate

this in some form into the prediction model. The aim of the computer package proposed here is to aid the analyst in the more mechanistic part of their job. However, if this is done in a closed system, it will be of very little use. Predictions should not be made solely on space time pattern analysis, and any method only allowing information of this sort to enter will be inadequate. Thus, in the statistical model, some means of entering subjective beliefs must also be incorporated.

Finally, if such a facility may be provided, a further refinement will be necessary. If human user's are to enter their own predictions for crime rates into the system, or modify those from the space-time models, some form of performance rating must be carried out, to weight the influence that these alterations may have. While it is reasonable to acknowledge there are some aspects of pattern analysis where the human analyst is better, to allow complete override could, with an over-enthusiastic user, simply lead back to the pre-analytical situation, with the human user's predictions being the only output. Therefore some decision should be made as to how much influence the subjective input should have, and this is most reasonably done in terms of past performance. In this way, at times when spatial pattern is the predominant characteristic, more weight will be given to the statistical model, but under more unusual circumstances (eg. sudden weather change, arrival of a carnival etc.) the human analyst is given control.

1.4 A Bayesian Approach

Given the objective of combining evidence in the data with a set of "prior" beliefs in a statistical framework, "classical" statistical models cannot be easily employed. These base inferential, estimative and predictive methodologies entirely on the analysis of the data, corresponding to the closed systems discussed in the previous sections. Instead of these the Bayesian approach is proposed (see for example Barnett, 1982). This is an alternative system of statistical inference, in which the analyst of a give data set supplies prior beliefs, and modifies these in the light of the data, into "posterior beliefs". Thus, information may be fed into the system by another channel than that of data collection.

Above, the Bayesian principle is stated in inferential terms; however a predictive interpretation can be made. From these "posterior beliefs" predictions may be made about future data values. This can further be extended to a dynamic scenario, where as data arrives, posterior beliefs are modified and updated predictions are made. Also, these posterior beliefs could be modified by the analyst if, at any time, they receive informal evidence that may cause them to modify their expectations for future events.

1.4.1 Statistical Methods

Most forecasting systems do not take geographical aspects of the model into account. They tend more to work entirely on time series based methodologies, such as exponential smoothing or Box Jenkins methods (Box and Jenkins, 1976). Since a major objective of the study here is

beat by beat crime prediction, spatial effects cannot be ignored. It is likely that, for example, a good indicator of forthcoming crime rates in a given beat will be based on present data not only for that beat, but on the surrounding beats crime rate data also.

In addition to this, for purposes of pattern identification, it is likely (see example) that some geographical patterns will take place on a much smaller scale even than this. Hence, techniques applied here should not only forecast on a beat by beat scale, but also be able to "flag" important developments on an individual scale, to warn of important patterns evolving. Techniques based, for example, on Knox testing (Knox 1964) on epidemiological technique for identifying epidemics of disease in space and time, might be applied to "spates" of crime, (say burglary) in a given area.

In this study it is proposed, firstly, to apply spatial statistical techniques to some crime data in order to identify the types of spatial, probabilistic models that could be used as the basis for crime prediction and pattern analysis, in space and time. After this task has been completed it then follows to build these models in the context of the Bayesian system described in the last section.

If this is attained, a system capable of using space-time modelling techniques to formulate predictions will have been developed. However, in addition to this, the system will be capable of combining the results of this type of pattern analysis with the extra information that may be supplied by the experienced analyst.

1.4.2 The Combination of Forecasts

In both of the previous sections, a need to combine the forecasts of both the human and statistical methods has been outlined. Fortunately, one has been proposed by Morris (Morris, 1974). Experts (man or machine) are required to state their beliefs about future crime rates, in the form of probability distributions (this is always the case in a Bayesian framework), and a mathematical framework for combining these is set out. The principal may be used in the context of combining predictive distributions from the spatial analysis model with those supplied by police users.

An added bonus of the approach given by Morris is that the prior beliefs supplied may be "calibrated" against past performance, so that, for example, the expectations of an expert with a tendency to underpredict crime rates would have their supplied figures shifted upwards to allow for this. In addition to this, more sophisticated calibrations may be performed, so that, for instance, the variance of past performance of expert advice can also be used as a criterion for weighted combination of man and machine forecasts.

An attempt will be made in this project to apply the theoretical framework put forward to a practical means of forecasting crime viewed as a geographical phenomena.

1.5 Software Design

As suggested in the previous sections, many statistical methods may be deployed in the prediction of crime forecasting and pattern analysis, and some of these will be of a high level of complexity. However, it is important to remember that the end users, although experts in the analysis of crime patterns, may not have the training to directly apply these statistical methods, or interpret their output. It is therefore essential that the computer software written for this system "hides" the internal statistical analyses, and presents results in terms of output more meaningful to police managers or crime pattern analysts. Also, bearing in mind the Bayesian approach of the system, police users will be required to feed in data about subjective beliefs. As mentioned previously, this should be supplied as a probability distribution: clearly, for an expert without statistical training, this is not reasonable. As with the output, the user must be asked for this information in a format they can easily understand.

Bearing both of these in mind, the prototype software package is to be designed in close communication with members of the Police Force. For example, a survey of police officer reports to different mapped output formats will be carried out. From this, the optimal types of map data display to be incorporated into the package may be discovered.

In addition to this, the software prototype must be designed so that if it is evaluated in a police environment, it will not be accidentally sent into

error status. An important factor in the success of a system of this type is ease of use, and the more secure a system is, in terms of "crash" avoidance, the more confident its users will be.

1.6 Evaluation

After the design of the forecasting software, and the operational software has been completed, the system as a whole will need to be evaluated. This evaluation should be carried out either on site, or, if this proves difficult to implement, at the research site, but with the co-operation of a police officer who will operate the system as though it were used on site. This police officer should have experience of crime pattern analysis.

The purposes of this analysis will be two-fold. Firstly, the accuracy of the forecasts will be monitored, and strong and weak points of the forecasting system will be identified. Secondly, the ease of use of the system will also be monitored, with comments from the police users. It is hoped that the second type of evaluation will identify areas in which the user interface could be improved, and also new methods of crime pattern analysis might be suggested for implementation into the system at a later date.

The aim of this study is an evaluation of the implementation of quantitative geographical techniques for crime pattern analysis. Clearly, part of this is to develop and evaluate a working prototype, but a second, equally important aspect is to suggest additions and

modifications to a successful prototype, in order to progress from this stage to a fully operational system.

1.7 Objectives

The main purpose of this PhD is to anticipate the availability of geographically referenced crime incidence data by developing the spatial forecasting methods in advance of crime database developments. The aim is to create and evaluate an automated crime pattern analysis system. As a system of this sort will be required to function on the basis of data that may be easily collated on site, emphasis will be placed on empirical means of pattern spotting, rather than on theories of criminology. Although the latter may provide insight into the processes leading to the committing of crimes, they do not directly identify notable quantitative traits in the data.

The forecasting system should be capable of integrating "intuitive" knowledge of police officers with forecasts of crime rates based on statistical techniques, to allow for events affecting crime rates that may not be detectable on past data alone. As the system will have to interact with the police officers in order to do this, careful design of the user-related input and output sections will be necessary. Indeed, throughout the study contact should be maintained with the Northumbria Police force, the intended "end users". It is also emphasised that this PhD is not meant to be a substantive investigation of any spatial patterns that may exist in crime incidence data for an arbitrary study region, instead it is concerned with more general methodological issues.

There has previously been a lack of in depth spatial statistical analysis of small area geographically referenced crime data, and this study intends to rectify this omission.

The system developed should provide the crime pattern analyst with a tool that will release their time currently spent on "mechanistic" pattern analysis, allowing them to apply more "humanistic" or intuitive skills to emerging patterns. One officer estimated that with non-analytical, database management software, 10% of the analyst's time is saved. With an analytical package of the type discussed here, this proportion will become about 40%. The extra 30% of time saved will allow the analyst to attain better subjective understanding of the patterns, and this may be fed back into the system via the Bayesian mechanism.

C H A P T E R 2

CRIME DATA : REQUIREMENTS AND CAPTURE TECHNIQUES

2.1 Introduction

In order to design and calibrate a crime pattern analysis system of the type proposed in the previous chapter, a clear idea of the structure of the readily available data on site is necessary. Further to this, if a prototype is to be constructed, a sample dataset will be needed. In this chapter, then, consideration will be given firstly to the nature of data that is suitable for crime pattern analysis, and also readily available to users of the system. After this, the methodology of data capture will be outlined. This methodology will exploit, in part, an anonymous manual crime report filing system already available in some subdivisions, to which the Northumbria Police force have allowed access for research purposes.

2.2 Sources of Crime-Related Data

The possible sources of data available to a crime pattern analysis system can be broadly divided into two main types: data relating directly to crime incidents, and data relating to variables that are thought likely to affect crime rates. In each case, methods of data recording will be considered, together with any related difficulties.

Subsequently, suitability of the data for incorporation into the methods proposed in this thesis will be evaluated. Obviously, data that is reliable is still of little use here if it cannot aid *substantially* in the prediction of crime rates on a week-by-week basis.

2.2.1 Data on Crime Incidence

On a large geographical scale (ie Police Force Divisions), yearly numbers of crimes are reported by the Home Office (Criminal Statistics in England and Wales Supplementary Tables Volume 3, from 1980 onwards). These are based on force wide figures, on a month-by-month basis. There are however obvious difficulties with this data source. The scale of *aggregation at which crime figures* are recorded in this secondary source is far larger than that required for this study. Clearly, if beat by beat, week by week predictions are to be made, and patterns identified at this scale of time and space, more detailed statistics over a smaller study area will be required. Thus, more local data sources must be considered, with a greater level of spatial identification.

On a more local scale it is likely that data compilation may have to be performed by the end user. Two main sources of such data would be by accessing records of prosecutions from Civil Court data or from police records of reported crimes. The first option is not suitable, since this data only relates to prosecution and so records of undetected crime will not appear. In addition to this, special formal channels of communication would have to be implemented, with

additional strain on the resources of both legal and police manpower. Clearly this would not be the case with the second option.

This leaves the option of collecting crime report data directly from the subdivision. This will provide data from a sufficiently local basis, and a framework for collecting information of some sort is already implemented in the crime clerk's office of subdivisions. A description of some type is required with every crime reported. From this, information about geographical location, date of event other details may be gained. In fact, certain subdivisions have manual filing and database systems implemented using the information from crime reports, which are used as simple crime pattern analysis tools.

For the needs of this pilot study, a subdivision which has employed a manual system such as that described above will provide a useful source of data. In addition, when a final computerised analysis system is completed, the data collection methods already implemented in the crime clerks office could be integrated into the data input procedure for the computer program without a great deal of modification.

Thus, in terms of data availability, and of geographical scope and resolution , the optimum information source for data directly relating to crimes would be those obtained directly at subdivision level. This argument applies to the final source of data on a fully operational system, and since the pilot study is intended to examine the feasibility

of such a system, must indeed also apply to the source of a test data set.

A further justification for this data source is that, in the future, a centralised, computerised crime incidence reporting system is envisaged, which will be able to supply electronic data of the type described above to subdivision. This is likely to contain similar information to that of the type currently manually reported in the crime clerks office.

2.2.2 Problems in the Collection of Crime Data

Consider crime related data from secondary sources, as set out above. The data will have been previously compiled in some format before being transferred into the database to be used for the pilot study. It is therefore relevant to give some thought to the initial recording process, and identify any problems associated with the database, and its contents, attempt to assess their effects, and suggest any remedial action that could be taken.

The major source of problems in crime databases relates to the non-reporting of some offences (Walker, 1983). Crimes will only become registered if they are either witnessed by a Police Officer, or reported by a civilian. There are several circumstances in which crimes would not be reported; Morrison, (1897) points out that crimes will often only be reported if they are actually perceived as crimes by

members of the public, or if members of the public approve of the legislation defining the criminal offences.

In more recent times Walker, (1983) has identified six major causes of non-reporting crime. These are now briefly outlined, and their effect will then be considered in a geographical context.

(1) No-one except the offender(s) are aware that the crime has occurred. This chiefly occurs in the cases of murder, or fraud.

(2) The victim is afraid of repercussion if they report the crime. This may be due to threats on the part of the offender.

(3) In the case of sex offences, the victim may be unwilling to give evidence to the Police or appear in court at a future date.

(4) The victim may feel that there is little the Police can do to help them, and may either not wish to appear in court or feel that reporting the crime would be a "waste of time". This may happen, for example in the case of household burglaries if the property is not insured, or alternatively if the victim feels there is little chance of detecting the offender in an assault or robbery case.

(5) The crime committed was considered to be trivial by the victim and it was not thought that it merited reporting. For example, theft of milk bottles from doorsteps.

(6) Among some immigrant communities, there is a *reluctance to* make contact with the Police. To quote McCulloch et al (1974) "There is evidence to indicate that many Asians do not take steps to have offences investigated due to fear of Police, difficulties of communication, mistrust of alien ways and ignorance".

In the first type of non-reporting, the frequency of types of crime to which this phenomenon is usually linked is low or the nature of the crime is not strongly geographically referenced so that crime pattern analysis of the type suggested in the first chapter is unlikely to be required. This is also the case with the second type of crime; it is not particularly easy to analyse crimes involving, say, blackmail using a system of this sort, firstly due to their rarity, and secondly due to the vagueness of their geographical referencing. It may be possible that a small proportion of the types of crime that could be easily analysed in a system of this type (the identification of which will be left to later in the Chapter), have their reporting censored in this way, but it is expected that the relative frequency of these to reported events will be small.

The major causes of censoring liable to have some geographical nature are those set out in examples 4 and 6. It is perhaps more likely that residents will have their properties insured, and are therefore more likely to report burglary offences. it is hoped, however, that although there may be greater likelihood of non-reporting in more deprived areas, this will be combated in some cases by the introduction of neighbourhood watch schemes. Again, in example 6,

there may be certain area having a higher concentration of immigrant communities, from which crimes are less likely to be reported than on average.

In both of these cases, the geographical aspect of the problem of non-reporting is that the under-representation of crime rates in various regions will not be uniform. Thus, the relative risks of some regions with respect to others may not be truly represented by analysis of this dataset. The effects discussed in the past paragraph may be confined to small neighbourhoods, so that the "loss rate" of crimes occurring but not being reported may vary within subdivisions. Thus there is a danger of some degree of distortion in the geographical patterns of crimes perceived by the analyst.

Perhaps this highlights the nature of the main problem of this thesis. Although in some regions, and perhaps over most time periods, the data will provide reasonable clues as to the possible future variation in potential crime patterns, future variation in potential crime patterns, there may be some aspects that data analysis is unable to pick up. Thus, a system requires some further means of input, perhaps from a human expert, which may be combined with the results of ordinary data analysis.

In addition to the distortions between crimes actually committed and those not reported, a further proportion may eventually be revealed not to be criminal offences. Perhaps for the purposes of this study this may not be too severe a problem. In the case of some incidents,

although no criminal offence is committed, there may be times when police officers are still called upon, for example to mediate in domestic disputes. If part of the purpose of the pattern analysis system is to forecast the manpower requirement for short-term horizons, then it is perhaps reasonable to incorporate a certain amount of non-criminal incidents, since these will certainly contribute to the total workload.

2.2.3 Data on Variables Which may Relate to Crime

In this section, data other than that directly relating to crime incidence, but which may be of use in the analysis of patterns or the prediction of future crime rates will be discussed. There will be two major headings here: Firstly, data concerning variables that are liable to be correlated with crime rates - possibly with some time-lagged effect, will be considered. If data of this sort may be collected on a week-by-week basis, this could be incorporated into a crime pattern analysis system, and used by prediction methods.The second type of data is that relating to local geography. This refers to digitised beat boundary outlines, together with various other cartographic detail. This will need to be converted to electronic format if the crime prediction system is to perform spatial modelling, or present mapped information on VDU to analysts.

Firstly consider those variables that could be correlated to crime rates. Obviously, many such variables could be speculated. For example, it may be possible to model crime risks in terms of numbers of potential offenders in the locality. This requires two assumptions:

firstly that journey-to-crime figures for most criminals relatively small, and secondly that some characteristics of a "typical" offender are known. In these cases, if the characteristics are based on say demographic or employment variables, an estimate of the number of people in whose category a larger number of "potential offenders" lie could be used as a means of assessing risk. Clearly, there are several tenuous links here; not all offenders are "typical" and a wide range of offender characteristics with a high degree of variance would mean that any "potential offender" category would be very large, in order to contain a reasonable proportion of the distribution of offenders. Furthermore, there is the "nearness to crime" assumption.

However, it is possible that, in a statistical sense, some variables of the form above may explain a reasonable percentage of variance. If this is the case, although in explanatory terms the analysis may not be particularly powerful, in predictive terms it may be of some use.

The main source of such data is that from the most recent population census. This will give counts of the population broken down by age, or sex or various other categories for census enumeration districts. These could relate for example to identifying age range or employment status or a combination of both of these associated with high risks of burglary. The enumeration districts, however, are an independent set of areal units to police foot beats. This implies that the census variables used in any analysis of this type would have to be estimated, by some means, for foot beats. The typical area of an urban ED is about 0.05 Km^2, and that of a foot beat is about 0.3 Km^2, so that

since areas are of similar magnitude, the proportion of EDs that are split over beat boundaries will not be negligible. Suppose some sort of allocation algorithm is employed, where if an ED is contained completely its entire population is attributed to the appropriate foot beat, but if it is straddled across beats, it is pro-rated to all relevant beats. Then, since a great proportion of beats are contained in the second, overflowing, category the implementation of the algorithm will be computationally expensive, and will also lead to fairly error-prone estimates (as the pro-rating is not an exact process).

In addition to this, the main purpose of the final crime prediction and analysis system is that of short-term forecasts. This implies that short-term crime phenomena of a few days span will need to be predicted. However, the census variables are only updated on a ten year basis. In comparison to the weekly updated crime count data, they will be virtually static. Also, in the ten year period, the neighbourhood or population characteristics that the census data attempts to measure may change drastically. New housing may be built, or areas may drop in affluence due, for example to the running down of a dominant local industry. Thus, the time-scale over which census variables are updated makes them infeasible for the task in hand.

On another level, they may be able to predict "base level" or average crime rates in foot beats, which one may expect to be relatively constant over time. However, given that the principal aim is prediction, rather than a causal analysis (which census data may not

aid with anyhow, due to problems in finding good proxy variables for any hypothesised processes) these base levels could more easily be calibrated by analysis of crime rates themselves on a beat by beat basis, over a long period of time. This would by-pass the areal unit reassignment problem, as crimes are systematically assigned to foot beats in the police force's data collection process. In addition to this, the burglary rate data will be more recent than census data; particularly for this study, where at the time of analysis the census is eight years behind the most recent crime data. Finally, in a working system, the crime data would be collated as a matter of course, whereas the census data would require extra resources, in terms of both cost and manpower, firstly to obtain the data, and then to process it, and present it in a form useful to the foot beat based system. Thus, due to the low frequency of update, and the inaccuracies of pro-rating data, census data would not seem viable in a working system.

The implications of this are that, while some type of exploratory analysis of census-based explanatory variables against crime rates as a response may be of interest to shed light on crime rates in an initial study, perhaps to relate crime rates to some variable known to cluster geographically, and hence gain some idea for modelling crime rates as a geographical process, they are not recommended for incorporation into a working system for the reasons stated above. Thus, any prototype system designed in this PhD will function on the basis of other variables than those from this source.

2.2.4 Cartographic Data

From the last section it would seem that there is little basis for using explanatory variable models to predict short term crime rates on an ongoing basis. The only variables that might be used for this purpose are the past crime rates themselves, as they are already collected at a suitable level of geographical resolution, and at a sufficiently high frequency. This would suggest that a spatial autoregressive model might be appropriate, since using recent levels in crime rates in neighbouring areas to predict those in a given area. In models of this sort, high levels of crime in some area are thought to be predictors of crime nearby, either due to the presence of the crime itself (ie. an offender may become familiar with an area, and return to it in the future) or due to some underlying phenomena that is also a process evolving in space and time. Note that the time sales discussed here are in the order of one or two weeks, not necessarily long term trends. Models of this type will be discussed in detail in Chapter 4. However, that it is clear that, since these methods involve considering crimes in the context of space , data relating to the nature of the space will be required in analysis, as well as that related to the crimes themselves. If crime rates are to be treated as a spatial process, nearness of foot beats to each other, and locations of population for example, may be important. In addition to this, digitised beat boundary information will be required in order to display mapped information on a VDU. Furthermore, it is required to assign spatially referenced crime cases to beats, using a point-in-polygon related

technique, and in order to do this, digital cartographic information relating to the beat boundaries will be needed.

Therefore, three main types of cartographic data will be required; a set of beat boundaries, which may be used for mapping purposes, a description of beat contiguity and a set of beat centroids of some description, which will be used as the basis for an autoregressive model, in which nearness between beats is a major factor in determining future rates.

The Northumbria Police force has copies of each subdivisional map of foot beat areas, drawn to a 1:10000 scale. These have been loaned to the author for research purposes, and for the use of this PhD, they may be digitised on university equipment, for whichever subdivisions are required for the purposes of this study. From these, centroids of beats, in a purely geometrical sense may be computed, using Geographical Information System (GIS) techniques. Similarly, the topology of the beats may be deduced. Thus, information will be available relating to the shape of a beat, which beats it is adjacent to, and how distant it is from other beats, based on a distance matrix between centroids. In addition to this, housing concentration centroids may also be obtained, since the maps supplied give locations of land plots, both private and commercial, in addition to beat boundaries. These maps also contain details of roads in the subdivision. Although not initially of use, it is possible that, at some future date, they may be incorporated into a mapping system.

All of this data, at least for one subdivision will be collected for the pilot study. However, in a final, operational system in police subdivisions there may be some need of a data updating and management system. This data would remain constant over reasonably long periods of time, although the building of new housing, new roads or the re-designation of foot beat boundaries may occasionally require some of the information to change. This would possibly be done at force level, with a periodic review of changes in local cartographic features, such as the roads on buildings mentioned above.

This should be relatively simple to implement, as police require up-to-date manual maps, and all the relevant information will be available in these. These situation may improve further still in coming years, with the adoption of Geographical Information System packages, which will store data of the type required electronically. This could then be transferred directly to crime pattern analysis software. It is likely that, with the advent of such equipment, map data "housekeeping" will be a formalised function of Force Headquarters, possibly by civilian staff, so that the burden of maintaining data will be shifted from the crime pattern analysis.

2.3 Data Set for Pilot Study : An Overview

At this stage, various sources of data have been identified, and evaluated for quality and relevance to the project under development. The major sources of data will be from subdivisional crime reports, in a dynamic sense, and in a more static background sense, from

digitised beat boundaries. Having considered these sources, selection of specific data required from these sources will now be considered in greater detail.

2.3.1 Current Manual Datasets : An Example

In this section some decisions must be made as to which data is to be collected. Certain types of crime will not be viable for analysis, either due to their rarity or due to their vagueness of geographical location (if, indeed, a geographical context exists at all). Even after identifying a suitable crime type, thought must be given to the particular recorded variables that will be of relevance to pattern analysis.

In order to do this, an example of a manual pattern analysis system will be considered. This will be of use in two ways. In the first case, it shows what function the crime pattern analyst has to perform, and which data is relevant to them. In the second instance, if this particular system is selected for examination, the actual data contained which will be used as a pilot crime data set, is set out in a format that facilitates easy, error free transcription.

2.3.2 The Databox System

This system was initially implemented in South Gosforth subdivision of the Northumbria police force. It was developed in the early 1980s by Inspector Jim Lillie. It was decided not to compile data on all

categories of crime, only those which patterns were thought to be relevant. The four classes of crime which were decided upon for analysis were:

(1) Burglary or Dwelling House

(2) Burglary of Other Building

(3) Theft of Motor Vehicles

(4) Theft from Motor Vehicles

For each of the above categories there is one box (this is literally the "Data Box"), and each box contains a folder for each police foot beat (There are 32 beats in South Gosforth). Within each of these folders, one sheet of paper is kept for each month. On that sheet is a matrix of crime records; the rows refer to individual crime reports, while the columns refer to attributes of each recorded case. These will be items such as time of day, location of event, date of event and several other items, based on items taken (as these are all based on thefts) and details of how the offence was committed.

The format of these sheets vary slightly between the four boxes, since details of importance are not the identical for all the types of crime, but an example sheet is reproduced in figure 2.1, for household burglary. There is information regarding the address of the offence, time of day, a crime reference number, point of entry was gained through a door, a window or another means. In addition to this, a list of items stolen is also included.

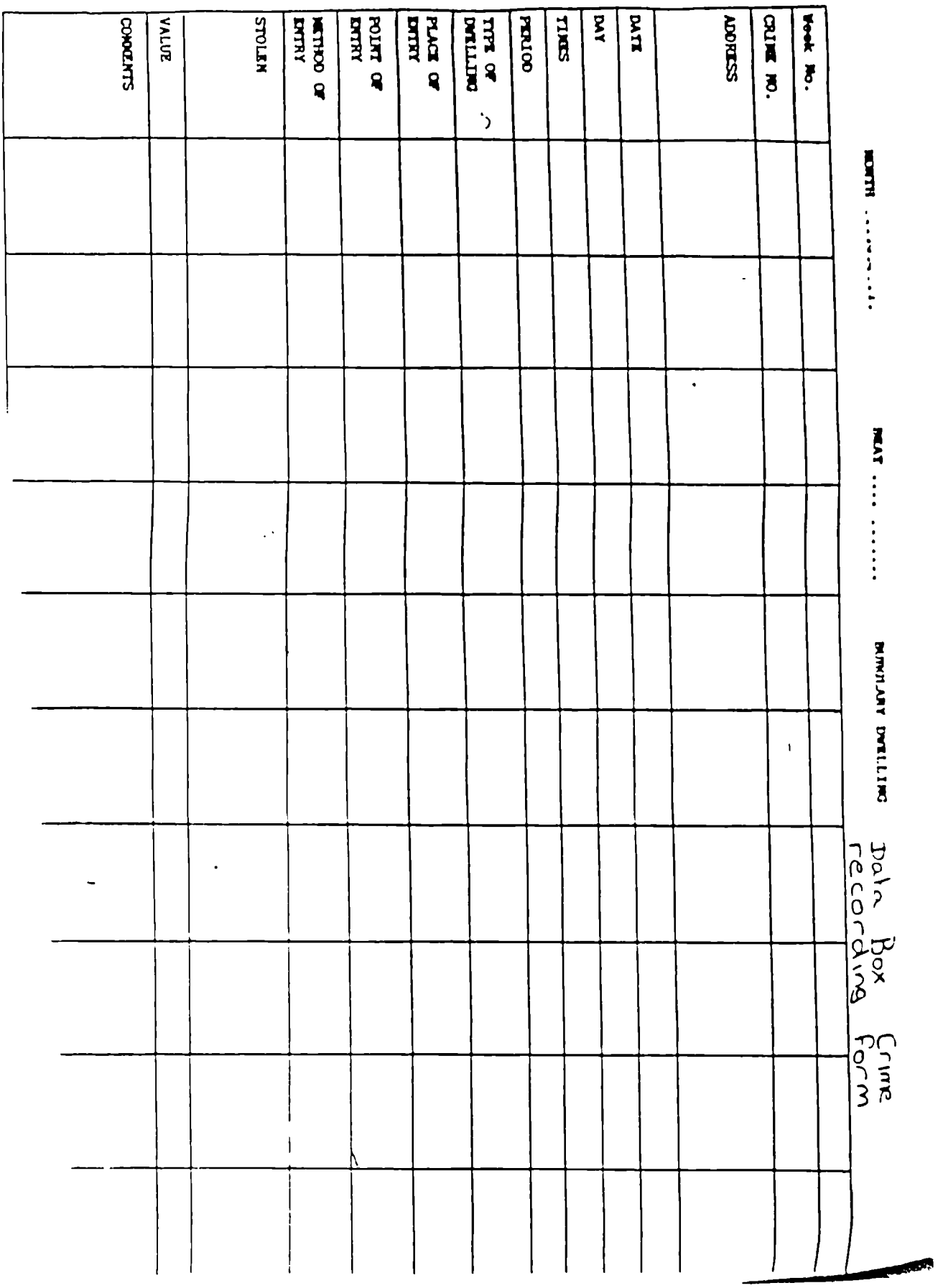

MONTH .....
BEAT .....
BURGLARY DWELLING
Data Box recording Crime form
Week No.
CRIME NO.
ADDRESS
DATE
DAY
TIMES
PERIOD
TYPE OF DWELLING
PLACE OF ENTRY
POINT OF ENTRY
METHOD OF ENTRY
STOLEN
VALUE
COMMENTS

Figure 2.1

The information is presented in the Databox system in order to allow the analyst to manually identify space- or time related patterns, and also to note any patterns in the methods used, or items taken by the offenders. The major aim here is to automate the detection of space and time patterns, although the information relating to methods used and items stolen may also be of interest, for example in categorising "typical" burglary types.

2.4 Selecting a Type of Crime for Analysis

Eventually, it is hoped that crime pattern analysis algorithms may be applied to all of the four categories of crime suggested by the "Data Box" system. However, it must be borne in mind that, in order to construct a pilot system, data must be transcribed from the manual system into electronic format. In addition to this, if techniques are developed for the analysis of a single type of crime, based on space-time pattern detection, it seems reasonable that, with only a small amount of modification, these may be extended to the other three types, since all of these may be thought of as phenomena constrained by space and time. It seems reasonable, therefore, to limit the pilot study to an analysis of a single crime type.

A reasonable type selection for this study is "Household Burglaries". The recording of the data for these is spatially well referenced (by an address), and there is little room for uncertainty in locating the event geographically. This may not be the case, for example, in a theft from a car where the owner (or even the offender) may be unable to

note the exact place at which theft occurred, since it was not noticed until some time later, during which the car had been moved six factors. Thus, the data set of crimes proposed for the pilot study will be, for a given period, the household burglary data from the "Data Box" system at South Gosforth subdivision.

2.5 Attaching Postcodes

On a typical "Data Box" record of a household burglary, the address of the victim's dwelling will be recorded. This is not, in itself, a useable format for a quantitative analysis technique. In order to map, and process this data, the location of the even must be supplied as a pair of coordinates. Thus, some method of converting verbal information into locational coordinates is required. This is a fairly difficult problem; the address, as a character string, must be matched on a look-up table of coordinates. However, there are several formats that the address may take. Firstly, there is the distinction between, say "St" and "Street" etc. In addition to this, sometimes addresses omit town names, or parts of names (ie. "South Newtown" becomes "Newtown". Because of this, if addressed are entered in an informal way, there will be a large proportion of unmatched strings. Because of the above effect, address to coordinate look-up systems would usually require a rigorously defined format for addresses to be entered. This would lead to difficulties in implementation.

A more viable alternative would be to use post codes . A look up table is available from the Post Office giving a coordinate pair (and

other information) for all postcodes in the UK (at time of release - the tape is regularly updated to remove postcodes of demolished areas, and to add new codes). Subsets of this lookup table relevant to the subdivision of interest could be stored in disc format on micros, and used to provide coordinate references for post-codes entered.

Postcodes are not accurate to the exact household, typically they contain about 15 or 16 houses. Thus, some compromise on specificity is made. However, if the postcode of an area is known, it can easily be converted to a format suitable for matching on a look-up table. Hence, in compensation for a slight loss in accuracy, a much larger proportion of crimes entered into the system will be matched to coordinates.

There are various other advantages in adopting a post-coded reference system. On a technical level, the storage overheads of a postcode-coordinate look-up table will be considerably less than that of an address-coordinate equivalent. This is because the size of a postcode string is only 8 characters, while the text of an address will greatly exceed this, and also since postcodes cover several houses, fewer postcode zones will cover a given region than house land plots.

In addition to this, data that is not spatially referenced to an individual household cannot be matched to a householder, so that anonymity is preserved. This data will then not require registration under the Data Protection Act (1984), which may otherwise prove

problematic. Thus, in some interpretations, the lack of specificity of postcodes may be seen as advantageous.

Also, currently large databases of varying kinds are being compiled that are geographically referenced. For example, Superprofiles (Charlton et al, 1985) a neighbourhood classification database. On the recommendation of the Chorley Report (Chorley 1987). Many of these are referenced using postcoding. Thus, crime data that is postcoded may be matched geographically to other datasets, using relational database technology of some point in the future. Thus, data compiled on crime incidence at a local scale, to be used initially as part of a crime pattern analysis system could later be passed on to research and development departments, for example, where its relationship with other neighbourhood referenced data could be studied.

Finally, in the Northumbria Police Force, who have agreed to supply pilot data, it is likely that in the next few years postcoding will be adopted on all crime reports, and that computerised records of all reports will exist, with postcodes included. Thus, if the pilot dataset includes postcode information the anticipated scenario in the near future will be simulated.

2.6 Data Capture

The problem of obtaining the data must now be addressed. The main issues are converting the format of data in the "Data Box" system

into a computer file format which may be analysed, the problem of post coding the address data, and finally that of means of transferring data from the data boxes in the subdivisional crime clerk's office onto the mainframe computer installation at the site of research.

2.6.1 Format for the Recording of Data

The data to be recorded is that presented in the "Data Box" corresponding to household burglary. However, certain aspects of the recording in the data box system are handled on an informal basis, whereas a more strict coding system will be required for this analysis. Certain items are required to be binary state variables (eg. is crime detected) and others are qualitative, requiring a categorical variable.

Finally, the items stolen will be encoded in list format, with code numbers corresponding to the type of item stolen. Eventually, this may be converted to a set of binary variables (ie. one variable corresponding to each type of article, with a "stolen"/"not stolen" indicator) however, it was felt by the author that for data coding, the first format was less likely to lead to transcription error, resembling the "Data Box" format more closely. It is possible that the binary format may be more convenient for analysis, but conversion to this format might be performed by computer at a later stage.

The proposed format of the prototype data set that is to be collected is given in table 2.1. This gives areas where data currently collectable may be stored, as well as areas where derived data will also be

Table 2.1

Format Of The Household Burglary Data Set

Variable Name	Column(s)	Description Of Variable
DAY	1	Day Of week Of Burglary 1=Sunday .. 7=Saturday
MONTH	2-3	Month of Burglary 1..12
DATE	4-5	Date Of Burglary
POSTCODE	6-13	Postcode Of Burglary
BEAT	14-15	Local Foot Beat Code
ME	16	Method of Entry F=Forced B=Break In I=Insecure Building D=Drilling O=Other
PE	17	Point of Entry D=Door W=Window O=Other
DE	18	Direction of Entry F=Front S=Side B=Back R=Roof O=Other

Table 2.1
Continued

Variable Name	Column(s)	Description Of Variable
STOLEN	19-24	Items Stolen Up to six out of C = Colour Tv H = Hi-Fi B = B/W Tv J = Jewellery V = Video Recorder E = Electrical Goods K = Cash F = Food X = Cheques D = Drink L = Clothes P = Personal Goods G = Furniture O = Ornaments M = Camera T = Tools Q = Other Or A = Attempt (No items Stolen)

stored. This derived data will consist of the results of running the look-up table software from postcodes to grid coordinates, and also of converting dates from a month-and-date format to a single day-of-year. These numbers are directly subtracted, and therefore more convenient for computer analysis.

Having decided to use postcoding to geographically represent the data, handwritten addresses in the "Data Box" system must be post-coded. This will be done using the Area Postcode Directory. Certain missing values may occur, either due to incorrect address recording, or due to burglaries occurring at address that did not exist at the last time of compilation of the directory. However, in the collected data set the proportion of these was about 8%, which did not cause major difficulties.

In compiling this dataset, going through addresses one-by-one and manually postcoding is clearly a time consuming exercise. However, it must be borne in mind that eventually this will be carried out as a matter of course as crimes are entered into the system, possibly with the victims being able to supply their own postal codes.

2.6.2 Data Transference

All of the data required for the pilot dataset is stored in the subdivisional crime clerk's office, and it is required to transfer this onto a computer file at the research site, but the original copy of the data must not leave the crime clerks office. Therefore, the only

means of collecting the data is by regular visits (say two mornings a week) to the subdivision, recording the data on site. Permission has been granted to do this by the Northumbria Police Force. However, the task may be speeded up considerably by employing a portable computer as the data capture tool. The pilot data file may be entered into the machine (in this case an Epson PX-80) using a text editor program, and then, using file transfer software available on the mainframe, this data may be transferred. Without this technology, data would have to be copied by hand from police records to paper, and this would then have to be entered into the mainframe computer. Thus, using the portable computer method described above, the data need only be entered once, whereas otherwise it has to be transcribed and then typed. In the manual case, then, roughly twice the resources would be required, and there are two sources of human error (in transcription and in typing).

2.6.3 Data Collected

The "Data Box" system has been operational in the study subdivision since 1984, and using this as source one years worth of detailed records will be captured. Any period of less than one year could lead to problems of seasonal bias in the study data set when carrying out various types of analysis; Thus, one years worth of data is to be collected, and the Northumbria Police Force have allocated the author office space within the subdivision over a one year period for this purpose. However, there may be some types of analysis that require

more than one years data; for example if seasonal effects are to be considered, the rates of crime for more than one year should be recorded if periodicity is to be checked. For periods of longer than one year, beat by beat crime counts will be tabulated on a weekly basis. This allows seasonal rates, and spatial modelling on a foot beat level of resolution, to be carried out over longer time spans than a single year. These tables will be compiled until January 1987; since this only involves the transcription of crosstabulations, this is to be done on an informal basis without the allocation of special office space.

2.7 Conclusions

The content of this chapter may best be summarised by the table 2.2 . This identifies each dataset that will be required for analysis, in terms of its contents, and the source of the data. The methods of collecting the data described here apply only to the pilot study. It is hoped that if the eventual crime prediction software becomes operational, the data collection will become formalised. The data here serves the purposes of exploring, evaluating and calibration of the types of models that might be used in such software eventually. At a later point in the study, when a working prototype is to be evaluated, some software for direct input of data to the system will be considered. The main purpose of the data here is to provide a basis for the following two chapters, whose subject matter will be the exploratory analysis of crime pattern data, and then the building of specific space-time models for the prediction of household burglaries. The data used has to be of a form likely to be generally available to all police

forces throughout the 1990s, and the pilot data gathered here meets this requirement.

CHAPTER 3

EXPLORATORY DATA ANALYSIS

3.1 Introduction

In the previous chapter, the methodology for gathering a pilot data set for the analysis of crime patterns was set down. Having obtained such a data set, the aim of this chapter is to perform various trial analyses on this data, and gain some insight into the uses that quantitative techniques may be put to analyse crime patterns. The consideration of crime as a spatial process is separated from this chapter, as it is thought to be an area of sufficient importance with respect to the aims of chapter 1 as to merit a chapter of its own. Thus, exploratory analysis of crime as a geographical pattern is incorporated in chapter 4. Here, certain other aspects of crime pattern analysis must also be considered, since although not directly connected with space, they may have some bearing on the final crime pattern analysis system. For example, analysis of characteristics of the method of burglary employed are clearly important to police officers attempting to infer facts about phenomena leading to a large incidence of crime in a particular area.

Other aspects, such as the seasonal variations in crime rate are also of importance in explaining local crime patterns, and it is possible that they may be of use in some prediction methodology.

Hence, before undertaking to synthesise a general spatial process model central to a crime prediction system, some preliminary investigation of the data is necessary. In this chapter, three such investigations are to be carried out. Firstly, as suggested above, seasonal aspects of the data will be analysed. After this, consideration will be given to the time of day of crimes. This information is incorporated in the data collected and is more conveniently available for crime reports - ie, all 999 calls - with a greater degree of reliability. However, it is important to investigate how useful this data is. The times are generally given as intervals - often because the witness is only able to specify the event in this way, and it is debatable whether they are of sufficient accuracy to incorporate in a prediction system. Even if they are not, they may be of interest in their own right. Finally, the subjective information about the burglaries will be examined. This relates to the methodologies used by offenders, and also to the items stolen. This type of information, although not essentially geographical, may be usefully combined with the spatial information by users of a crime pattern analysis system, as suggested previously.

3.2 Time of Day of Crime Incidence

An important aspect of patterns of crime is the time of day of incidence. At certain times of day, particular types of crime may be more likely to occur, and if police resource managers are aware of information of this type, appropriate allocation of reserve manpower can be made. In addition to this, it will provide officers on foot or car beat patrol with useful information. It is expected that the distribution of crime rates

over the day will vary between types of crime, and may be unique to different geographical areas. For example, in terms of "defensible space" models, (Newman, 1972) certain areas within neighbourhoods may become less visible as night falls. This may provide better opportunities for household burglary, and possibly cause the likelihood of such events to rise in the locality. The time of day that such an event may happen would depend on local architecture, and on the time of day of sunset. Both of these factors will vary geographically, the second also having a seasonal component.

Thus, it may not be particularly helpful to consider the distribution of crimes throughout the day on a national basis. It may, however, be possible to consider data collected locally, and, at subdivisional level evaluate intra-daily changes in crime risk. This could be done for all data aggregated over the entire subdivision, or broken down by individual foot beats. The first option, although less informative, may be a more suitable compromise, since the scarcity of numbers of incidents for individual beats may lead to problems in estimating distributions.

It may also be of interest to compare the patterns observed during working days to those at weekends. This may affect the times of day that potential criminals come into contact with opportunities of committing offences. Thus, a local analysis of time of day of crime risks taking into account the covariates of season, foot beat area and time of week may provide useful information for police resource administration and for direct policing.

In line with the rest of this PhD, a pilot study here will be carried out on household burglaries. The arguments presented in chapter 2 concerning the well defined geographical referencing of household burglaries, and the fact that they provide a reasonably large database apply equally here as in the other studies.

Thus, the problem here may be stated in the following stages:

i) Methodology specification - data collection
- analysis technique

ii) Data Evaluation

iii) Data Analysis

iv) Conclusions

The data evaluation will be interrelated to the analysis. As suggested previously, the level of sophistication of the data analysis will depend on the amount, and the quality, of the data supplied. The conclusions should be considered in terms of the feasibility of installing this type of analysis at a subdivisional level throughout the entire force, as well as in respect of interpreting the local patterns observed.

3.2.1 Data Collection

There is an initial problem in the non-reporting of certain crimes (Walker, 1983 or Sparks et al, 1977). Not all household burglaries occurring will be reported to the police, and police records of call-outs provide the most plentiful and easily obtainable database of the sort

required for this study. It is hoped, however, that generally this will not effect conclusions. Particularly, it will be assumed that the risk of a given crime not being reported is uniform throughout the day.

The major problem with data of this type is that the occupiers of houses are often out when household burglaries occur. Thus, the exact time of burglary is unknown. This may also be the case if the occupiers are in, but are unaware of a theft taking place (for example, during the night). Thus, rather than an exact time of burglary being known, an upper and lower limit is the maximum information that is available to the victim, or to the police. This limitation has been considered by the police; when recording data of this type an upper and lower boundary on time of event must always be entered, even when the crime was witnessed. In the latter case, a one or two minute gap is entered.

Hence, the database is to be compiled from these records, noting upper and lower limits of time of incident, and also calendar information, giving day of week, and week in year. In order to compare each season, an entire year of day is the minimum requirement. Given the manual nature of data collection in this study (the crime incidence records in which the time of day is noted are not yet computerised) the pilot study will be restricted to a single year. This data will then be coded and analysed statistically by computer.

Unfortunately, at this stage, information relating to foot beat areas is not stored on record, so that this part of the analysis may not be performed in the pilot study.

3.2.2 Method of Analysis

In the last section, the special nature of the data was considered. There is no point indicator of time of day available, only an internal estimate. For point estimates, there are many methods of obtaining models of probability distributions. In crudest form, simple histograms could be constructed, say on an hour-by-hour basis, giving direct visual information about the distribution of risk throughout the day.

On a more sophisticated level, Kernel estimation techniques (chapter 3 and Silverman, 1976) could be used to gain an approximated probability density function for daily crime variation, or parametric models of cyclic pattern in variation, (such as Fourier analysis) could be used. However, in all of these approaches, the introduction of internal data gives rise to technical complications, some of which have not yet been addressed.

Statistical aspects of distribution estimation will now be considered, with the special case of interval data. Firstly consider a maximum likelihood, non-parametric approach. In this case estimates of the cumulative distribution must be made of an unspecified function F, (which may or may not be continuous) but this cannot be gathered from a finite sample of intervals; however, F may be estimated for a finite set of ordinates. Suppose the sample of upper and lower intervals is (x_{li}, x_{ui}) $i = 1,n$. Then $\hat{F}(x_{li})$, $\hat{F}(x_{ui})$ may be considered. Denote these F_{ui}, F_{li}. The probability of x falling between x_{li} and x_{ui} is $\hat{F}_{ui} - \hat{F}_{li}$. Thus, the likelihood of the entire dataset is given by

$$L(F|\{x_{ui}, x_{li}\}) = \prod_{i=1}^{n} (F_{ui} - F_{li})$$

Also, as $\hat{F}$ is a cumulative distribution function, $1 \geq F(x) \geq F(y) \geq 0$ if $x \geq y$. Thus, to maximise $L(F|\{x_i, x_i\})$, the problem is identical to

$$\max \prod_{i=1}^{n} (F_{ui} - F_{Li})$$

subject to

$$F(x) \geq F(y)$$

(subject to $x \geq y$)

although this can be solved, the solution is not always easy to interpret. Suppose the set of F_{Li}'s and F_{ui}'s are ordered according to corresponding X_{Li}'s and X_{ui}'s. Then, the lowest element will be X_{Li}. Clearly, this must equal zero, otherwise $(F_{u2} - F_{L1})$ will not be maximised, if all other X's are given. Also, however, if $F_{L2} > F_{u1}$, F_{L2} must also be zero, maximising $F_{u2} - F_{L2}$. If the next highest ordinate is also an upper limit, F_{ui}, then, given the constraint $F_{ui} \geq F_{u1}$, and the fact that F_{u1} must maximise $(F_{u2} - F_{L1})$, we have $F_{u1} = F_{ui}$.

Thus, pr $(x \in (F_{u1}, F_{ui})) = 0$. Therefore, often, the maximum likelihood density estimate has regions of zero probability. In an interactive appreciation of the substantive issue, this does not seem satisfactory.

Parametric approaches are problematic in a different sense; in this case, the density is modelled as $f(\underline{x}; \underline{\theta})$, and a maximum likelihood estimate of is given by

$$\max_{\underline{\theta}} \prod_{i=1}^{n} \int_{x_{Li}}^{x_{ui}} f(x; \theta)\, dx$$

often the optimisation problem becomes computationally expensive, particularly if the integral may not be expressed algebraically for some given f.

This leaves the option of a modification of the Kernel estimation technique which may be applied to interval estimates. A reasonable approach may be to use the method of Kernel estimation for point data, based in the centres of intervals, and widening the bandwidths of the Kernel distributions in proportion to the size of interval. Thus, for example,

$$f_K(x) = \frac{1}{n} \sum_i g\left(\frac{x - (x_{ui} - x_{li})/2}{K(x_{ui} - x_{li})} \right)$$

the simplest of these would be

$$f_K(x) = \frac{1}{n} \sum_i g_i(x)$$

where

$$g_i(x) = (x_{ui} - x_{li})^{-1}$$

if $x \in (x_{li}, x_{ui})$

$$= 0$$

otherwise (1)

(1) is the equivalent to assigning a rectangular distribution between upper and lower limits, for each pair, and then averaging over all of these . Assuming these rectangular distributions hold, or at least represent victim knowledge of the time of crime, the resultant average distribution of the time of day of an event generated first by selecting a

past record at random, and then assigning time of day according to the rectangular distribution corresponding to that choice.

This method, though theoretically less appealing than the others suggested has the advantage of being relatively easy to calculate, and also of not having "pockets" of zero probability. It also directly synthesises a density function, rather than a cumulative function for the approximate distribution. Thus, the resultant may be easily interpreted as a "risk profile" of household burglaries throughout the day.

A further requirement has to be made for analysis in terms of time of day: for intervals including midnight, a modulo-based interval estimator must be used, so that g_i is added from $\mathcal{T}_{L_i}$ to midnight, and then from midnight to $\mathcal{T}_{U_i}$. This is particularly important in analyses where the day of week is taken into account.

Another problem is that of extremely large intervals. In certain cases, for example when the occupiers of the burgled property have been away for several days on holiday, there is an extremely long interval between the upper and lower time boundaries. In terms of the day-profile, the rectangular distribution is virtually uniform throughout the how period. This has the effect of "flattening" the average distribution. Effectively, there are two main factors in the data that effect the interval-based estimator. Firstly, centres of intervals tend to build up a histogram-based model of the distribution, but this interacts with the interval size. The peaks that centroids may produce will be "smoothed" by larger intervals. When the interval size gets large, the smoothing

effect affects the entire distribution estimate. It may therefore be appropriate to "downweight" contributions of rectangular distributions with larger interval sizes. It seems reasonable, furthermore, that any intervals exceeding 24 hours should be zero weighted.

Thus, the final formula may appear as

$$f_K(x) = \frac{\sum_i [w(x_{ui} - \bar{x})/(x_{ui} - x_{li})]}{\sum_i w_i(x_{ui} - x_{li})}$$

$$(\bar{x} = (x_{ui} + x_{li})/2)$$

where w is a weighting function, such that

$w(24) = 0$ if x_{li}, x_{ui} calibrated in hours

$w(x)$ is monotone decreasing if $x > 0$

The simplest $w(x)$ would be

$w(x) = 1$ if $x < K$

$= 0$ otherwise.

This simply cuts out all observations when the interval exceeds some given value. For the pilot study, this method will be employed.

3.2.2 Data Evaluation

In the last section, a method of estimating the distribution of household burglary occurrences throughout the day was proposed. This method may be applied to interval data concerning the time of day of household

burglaries. It was also suggested that those pairs of limits spanning beyond some reasonable amount of time be excluded. It will be an important problem to determine a suitable exclusion limit. In the interests of avoiding "over-smoothing" the limit must not be too high. However, exclusion of too many cases leads to reduction in sample size, and this will lead to "over-spikiness" in the estimates, as they will be based on only a few observations. It is therefore important to examine the numbers of records, and also the distribution of interval lengths in these records. There are 2025 observations in total.

The upper tail of interval distribution shows a slight peak, corresponding to those offences that occurred while the property was unoccupied for several days (or possibly several weeks). The effect of the upper tail is that the distribution mean exceeds the median. In the context of choosing cut-off points, the median and other quantiles are of more importance than moments of the distribution, since they allow a direct link to be made between proportions of cases lost and the interval lengths.

The median here is 6 hours, thus a cut off point of k=6 (see last section) would result in exclusion of 50% of cases, that is 1012 cases. Although removal of 50% of cases may seem large, it must be recalled that intervals of much larger times than this will distort the reasonable information given by the data in the lower half of the distribution. However, for kernel-based techniques, 1012 cases will still provide a good estimate of the generating distribution. Many studies have been based on considerably smaller samples than this.

The second issue here concerns the splitting of the data by a week/weekend division, and a seasonal division. Clearly, in the case of considering the data as a whole, a cut-off interval of 6 hours is appropriate. However, if the data is sub-divided, so that separate distribution estimates are considered for each division, then each estimate will be based on a smaller number of observations, with the corresponding loss of reliability.

It is possible that one data analysis may be performed when splitting by season, and another by week/weekend division, allowing the two effects to be assessed independently, but that splitting by both in the same analysis, allowing interaction to be considered, will require too many subdivisions, leading to an unacceptable reduction in the number of observations in each class.

Frequency counts are shown in table 3.1, giving counts and average lengths of intervals in each subcategory. Firstly, data are categorised for single data points. Thus, this allows checking of feasibility for independent analysis. After this, if the above is feasible, it may then be possible to carry out an interaction effect analysis. To test for this the crosstabulation may be analysed.

In the case of the full crosstabulation, if the cutoff of 6 hours is still applied, the smallest category contains 48 cases. This, although giving patchy coverage in some of the smaller data sets, may still be reasonable for examining risk profiles, particularly if those results gained from very small data sets are viewed cautiously.

Table 3.1
Occurence of crime by Time Of Year and Time Of week

	Winter	Spring	Summer	Winter	Total
Week	236	147	157	176	716
	2.292	2.214	2.029	2.497	2.269
Weekend	68	55	48	71	242
	2.728	2.555	2.615	2.296	2.539
Total	304	202	205	247	958
	2.390	2.307	2.166	2.429	2.439

(Upper figures denote incedent counts, Lower figure the average uncertainty in time of day of occurrence.)

When considering divisions by each factor in turn, there are reasonably sized samples for each category, so that, although interaction analysis of the two factors may not be reasonable, in all cases, separate seasonal and week/weekend analyses will be feasible.

3.2.4 Results of Analysis

A computer program to carry out the analysis proposed in the second section was written (in Prospero FORTRAN 77 running under MS-DOS). This is shown in listing 3.1. A minor change from the theoretical method has been made, in that a day has been divided into 48 half hourly intervals, rather than treated as continuous time. This speeds up computation and aids on-screen graphical representation.

The results for all incidents (table 3.2) incidents divided by week/weekend (table 3.3), divided by season (table 3.4) and by both week/weekend and season (table 3.5) are listed. Corresponding graphical representations are given in figures 3.1-3.4.

The problems of small samples are reflected in the "patchiness" of the graphs obtained from some of the two-way split datasets. However, these reflect to some extent patterns which may be more strongly identified in the single dimensional splits.

Firstly, consider the week/weekend splitting. On weekdays, risk peaks in the early to mid afternoon and again in the evening, at around 21.00 hours. A lower peak also occurs at about 2.00 hours. The least risk

Table 3.2

Time Of Day Distribution Of Household Burglaries : All Times

Time	Percent	Time	Percent	Time	Percent
00:00	1.71%	00:30	1.64%	01:00	1.56%
01:30	1.92%	02:00	2.32%	02:30	1.34%
03:00	2.11%	03:30	1.70%	04:00	1.23%
04:30	1.20%	05:00	0.84%	05:30	0.68%
06:00	0.66%	06:30	0.87%	07:00	0.70%
07:30	0.64%	08:00	0.50%	08:30	0.66%
09:00	0.84%	09:30	1.13%	10:00	1.15%
10:30	1.58%	11:00	1.95%	11:30	1.85%
12:00	2.13%	12:30	2.69%	13:00	2.26%
13:30	2.74%	14:00	3.00%	14:30	3.42%
15:00	3.59%	15:30	3.18%	16:00	2.93%
16:30	2.77%	17:00	2.01%	17:30	1.72%
18:00	1.94%	18:30	2.44%	19:00	2.92%
19:30	3.53%	20:00	4.12%	20:30	4.17%
21:00	3.67%	21:30	3.62%	22:00	3.52%
22:30	2.68%	23:00	2.29%	23:30	1.88%

Average gap = 2.337 Hrs.
Events used = 958

Table 3.3

Time Of Day Distribution Of Household Burglaries

Weekdays

Time	Percent	Time	Percent	Time	Percent
00:00	1.54%	00:30	1.15%	01:00	1.06%
01:30	1.56%	02:00	2.33%	02:30	1.27%
03:00	2.21%	03:30	1.37%	04:00	1.04%
04:30	1.04%	05:00	0.72%	05:30	0.72%
06:00	0.68%	06:30	0.81%	07:00	0.70%
07:30	0.64%	08:00	0.50%	08:30	0.71%
09:00	0.96%	09:30	1.24%	10:00	1.27%
10:30	1.84%	11:00	2.27%	11:30	2.10%
12:00	2.50%	12:30	3.04%	13:00	2.62%
13:30	3.43%	14:00	3.56%	14:30	4.12%
15:00	4.35%	15:30	3.79%	16:00	3.36%
16:30	2.97%	17:00	1.97%	17:30	1.72%
18:00	1.81%	18:30	2.43%	19:00	2.81%
19:30	3.32%	20:00	3.90%	20:30	3.70%
21:00	3.32%	21:30	3.01%	22:00	3.06%
22:30	2.11%	23:00	1.78%	23:30	1.56%

Average gap = 2.269 Hrs.
Items used = 716

Weekends

Time	Percent	Time	Percent	Time	Percent
00:00	2.22%	00:30	3.08%	01:00	3.06%
01:30	2.98%	02:00	2.28%	02:30	1.55%
03:00	1.81%	03:30	2.69%	04:00	1.80%
04:30	1.68%	05:00	1.21%	05:30	0.57%
06:00	0.61%	06:30	1.02%	07:00	0.68%
07:30	0.62%	08:00	0.48%	08:30	0.50%
09:00	0.48%	09:30	0.81%	10:00	0.77%
10:30	0.82%	11:00	1.01%	11:30	1.12%
12:00	1.01%	12:30	1.67%	13:00	1.19%
13:30	0.70%	14:00	1.31%	14:30	1.36%
15:00	1.34%	15:30	1.38%	16:00	1.66%
16:30	2.18%	17:00	2.11%	17:30	1.70%
18:00	2.30%	18:30	2.48%	19:00	3.24%
19:30	4.14%	20:00	4.77%	20:30	5.57%
21:00	4.72%	21:30	5.46%	22:00	4.88%
22:30	4.39%	23:00	3.78%	23:30	2.80%

Average gap = 2.539 Hrs.
Items used = 242

Table 3.4
Time Of Day Distribution Of Household Burglaries

Winter

Time	Percent	Time	Percent	Time	Percent
00:00	1.91%	00:30	1.27%	01:00	0.69%
01:30	1.52%	02:00	1.61%	02:30	0.70%
03:00	1.44%	03:30	0.99%	04:00	0.73%
04:30	1.02%	05:00	0.71%	05:30	0.93%
06:00	0.77%	06:30	0.93%	07:00	0.72%
07:30	0.35%	08:00	0.43%	08:30	0.65%
09:00	1.06%	09:30	0.89%	10:00	1.49%
10:30	1.37%	11:00	2.40%	11:30	2.24%
12:00	2.30%	12:30	2.21%	13:00	2.71%
13:30	2.28%	14:00	2.72%	14:30	2.69%
15:00	3.54%	15:30	2.68%	16:00	3.37%
16:30	3.74%	17:00	2.56%	17:30	2.66%
18:00	2.40%	18:30	3.10%	19:00	4.29%
19:30	4.29%	20:00	4.42%	20:30	3.91%
21:00	3.68%	21:30	3.82%	22:00	3.30%
22:30	2.90%	23:00	1.96%	23:30	1.66%

Average gap = 2.390 Hrs.
Items used = 304

Spring

Time	Percent	Time	Percent	Time	Percent
00:00	1.33%	00:30	1.74%	01:00	0.65%
01:30	1.69%	02:00	1.54%	02:30	1.36%
03:00	2.84%	03:30	2.05%	04:00	1.03%
04:30	1.25%	05:00	1.24%	05:30	0.76%
06:00	0.71%	06:30	0.96%	07:00	1.16%
07:30	1.08%	08:00	0.49%	08:30	0.45%
09:00	0.66%	09:30	1.92%	10:00	0.89%
10:30	1.61%	11:00	0.62%	11:30	0.95%
12:00	2.33%	12:30	3.01%	13:00	1.69%
13:30	3.53%	14:00	3.29%	14:30	3.58%
15:00	3.86%	15:30	3.36%	16:00	2.27%
16:30	2.30%	17:00	1.40%	17:30	1.21%
18:00	1.48%	18:30	1.60%	19:00	2.28%
19:30	4.07%	20:00	4.64%	20:30	5.08%
21:00	4.64%	21:30	3.63%	22:00	3.86%
22:30	3.17%	23:00	2.67%	23:30	2.08%

Average gap = 2.307 Hrs.
Items used = 202

Table 3.4 (continued)
Time Of Day Distribution Of Household Burglaries

Summer

Time	Percent	Time	Percent	Time	Percent
00:00	1.70%	00:30	2.33%	01:00	3.07%
01:30	3.06%	02:00	3.30%	02:30	1.55%
03:00	3.15%	03:30	1.38%	04:00	1.33%
04:30	2.03%	05:00	0.94%	05:30	0.50%
06:00	0.49%	06:30	0.46%	07:00	0.47%
07:30	0.80%	08:00	0.69%	08:30	1.09%
09:00	1.01%	09:30	1.40%	10:00	0.80%
10:30	1.67%	11:00	3.11%	11:30	2.18%
12:00	2.38%	12:30	3.30%	13:00	2.14%
13:30	3.03%	14:00	2.77%	14:30	4.71%
15:00	3.53%	15:30	2.90%	16:00	2.97%
16:30	1.83%	17:00	1.44%	17:30	0.93%
18:00	1.15%	18:30	1.76%	19:00	1.39%
19:30	1.45%	20:00	2.74%	20:30	3.30%
21:00	2.98%	21:30	3.65%	22:00	3.42%
22:30	2.44%	23:00	3.05%	23:30	2.25%

Average gap = 2.166 Hrs.
Items used = 205

Autumn

Time	Percent	Time	Percent	Time	Percent
00:00	1.79%	00:30	1.44%	01:00	2.14%
01:30	1.65%	02:00	3.00%	02:30	1.94%
03:00	1.47%	03:30	2.56%	04:00	1.95%
04:30	0.70%	05:00	0.59%	05:30	0.46%
06:00	0.64%	06:30	1.04%	07:00	0.49%
07:30	0.49%	08:00	0.42%	08:30	0.49%
09:00	0.58%	09:30	0.57%	10:00	1.23%
10:30	1.73%	11:00	1.54%	11:30	1.85%
12:00	1.53%	12:30	2.52%	13:00	2.27%
13:30	2.42%	14:00	3.28%	14:30	3.11%
15:00	3.48%	15:30	3.88%	16:00	2.90%
16:30	2.75%	17:00	2.29%	17:30	1.64%
18:00	2.39%	18:30	2.88%	19:00	3.02%
19:30	3.88%	20:00	4.46%	20:30	4.48%
21:00	3.44%	21:30	3.36%	22:00	3.60%
22:30	2.22%	23:00	1.74%	23:30	1.68%

Average gap = 2.439 Hrs.
Items used = 247

Table 3.5
Time Of Day Distribution Of Household Burglaries
Weekdays Winter

00:00	1.56%	00:30	0.86%	01:00	0.60%
01:30	0.91%	02:00	1.45%	02:30	0.73%
03:00	1.36%	03:30	0.78%	04:00	0.86%
04:30	0.86%	05:00	0.86%	05:30	1.07%
06:00	0.86%	06:30	1.07%	07:00	0.79%
07:30	0.32%	08:00	0.43%	08:30	0.61%
09:00	1.14%	09:30	0.92%	10:00	1.77%
10:30	1.61%	11:00	2.78%	11:30	2.47%
12:00	2.55%	12:30	2.54%	13:00	3.18%
13:30	2.70%	14:00	2.84%	14:30	3.18%
15:00	4.24%	15:30	3.07%	16:00	3.93%
16:30	3.89%	17:00	2.72%	17:30	2.71%
18:00	2.29%	18:30	3.14%	19:00	3.66%
19:30	3.86%	20:00	4.00%	20:30	3.72%
21:00	3.37%	21:30	3.56%	22:00	2.93%
22:30	2.55%	23:00	1.33%	23:30	1.40%

Average gap = 2.292 Hrs.
Items used = 236

Weekends Winter

00:00	3.13%	00:30	2.70%	01:00	0.98%
01:30	3.66%	02:00	2.19%	02:30	0.59%
03:00	1.73%	03:30	1.73%	04:00	0.26%
04:30	1.60%	05:00	0.21%	05:30	0.46%
06:00	0.46%	06:30	0.46%	07:00	0.46%
07:30	0.46%	08:00	0.46%	08:30	0.78%
09:00	0.78%	09:30	0.78%	10:00	0.50%
10:30	0.50%	11:00	1.07%	11:30	1.44%
12:00	1.44%	12:30	1.07%	13:00	1.09%
13:30	0.83%	14:00	2.30%	14:30	1.00%
15:00	1.12%	15:30	1.33%	16:00	1.41%
16:30	3.25%	17:00	2.03%	17:30	2.46%
18:00	2.80%	18:30	2.95%	19:00	6.47%
19:30	5.78%	20:00	5.90%	20:30	4.58%
21:00	4.74%	21:30	4.74%	22:00	4.56%
22:30	4.10%	23:00	4.13%	23:30	2.55%

Average gap = 2.728 Hrs.
Items used = 68

Figure 3.5 (continued)
Time Of Day Distribution Of Household Burglaries

Weekdays Spring

00:00	1.21%	00:30	1.75%	01:00	0.33%
01:30	1.83%	02:00	1.08%	02:30	1.33%
03:00	2.69%	03:30	1.39%	04:00	0.56%
04:30	0.70%	05:00	0.69%	05:30	0.69%
06:00	0.69%	06:30	1.03%	07:00	1.30%
07:30	1.30%	08:00	0.61%	08:30	0.62%
09:00	0.84%	09:30	1.90%	10:00	0.49%
10:30	1.48%	11:00	0.80%	11:30	1.11%
12:00	2.60%	12:30	3.47%	13:00	1.93%
13:30	4.45%	14:00	4.26%	14:30	4.21%
15:00	4.65%	15:30	4.03%	16:00	2.54%
16:30	2.48%	17:00	1.24%	17:30	1.04%
18:00	1.37%	18:30	1.70%	19:00	2.73%
19:30	3.87%	20:00	4.92%	20:30	4.22%
21:00	4.54%	21:30	3.11%	22:00	3.50%
22:30	2.66%	23:00	2.29%	23:30	1.80%

Average gap = 2.214 Hrs.
Items used = 147

Weekends Spring

00:00	1.66%	00:30	1.73%	01:00	1.50%
01:30	1.34%	02:00	2.79%	02:30	1.43%
03:00	3.25%	03:30	3.84%	04:00	2.28%
04:30	2.73%	05:00	2.73%	05:30	0.98%
06:00	0.77%	06:30	0.77%	07:00	0.77%
07:30	0.52%	08:00	0.15%	08:30	0.00%
09:00	0.15%	09:30	1.97%	10:00	1.97%
10:30	1.97%	11:00	0.15%	11:30	0.52%
12:00	1.63%	12:30	1.78%	13:00	1.05%
13:30	1.05%	14:00	0.69%	14:30	1.90%
15:00	1.75%	15:30	1.55%	16:00	1.55%
16:30	1.81%	17:00	1.81%	17:30	1.65%
18:00	1.77%	18:30	1.35%	19:00	1.05%
19:30	4.58%	20:00	3.88%	20:30	7.39%
21:00	4.89%	21:30	5.03%	22:00	4.83%
22:30	4.53%	23:00	3.67%	23:30	2.83%

Average gap = 2.555 Hrs.
Items used = 55

Figure 3.5 (continued)
Time Of Day Distribution Of Household Burglaries

Weekdays Summer

00:00	1.62%	00:30	1.56%	01:00	1.93%
01:30	2.84%	02:00	3.79%	02:30	0.93%
03:00	3.66%	03:30	0.72%	04:00	1.36%
04:30	1.93%	05:00	0.57%	05:30	0.32%
06:00	0.38%	06:30	0.34%	07:00	0.23%
07:30	0.67%	08:00	0.62%	08:30	1.20%
09:00	1.10%	09:30	1.74%	10:00	0.95%
10:30	2.19%	11:00	3.42%	11:30	2.21%
12:00	3.02%	12:30	4.22%	13:00	2.71%
13:30	3.87%	14:00	3.52%	14:30	5.99%
15:00	4.32%	15:30	3.58%	16:00	3.52%
16:30	1.97%	17:00	1.45%	17:30	0.92%
18:00	1.16%	18:30	1.18%	19:00	1.12%
19:30	1.07%	20:00	2.60%	20:30	3.38%
21:00	2.58%	21:30	2.69%	22:00	2.38%
22:30	2.13%	23:00	2.36%	23:30	1.96%

Average gap = 2.029 Hrs.
Items used = 157

Weekends Summer

00:00	1.97%	00:30	4.83%	01:00	6.83%
01:30	3.76%	02:00	1.68%	02:30	3.57%
03:00	1.49%	03:30	3.52%	04:00	1.23%
04:30	2.36%	05:00	2.13%	05:30	1.08%
06:00	0.85%	06:30	0.85%	07:00	1.22%
07:30	1.22%	08:00	0.92%	08:30	0.71%
09:00	0.71%	09:30	0.30%	10:00	0.30%
10:30	0.00%	11:00	2.08%	11:30	2.08%
12:00	0.30%	12:30	0.30%	13:00	0.30%
13:30	0.30%	14:00	0.30%	14:30	0.53%
15:00	0.95%	15:30	0.65%	16:00	1.17%
16:30	1.40%	17:00	1.40%	17:30	0.98%
18:00	1.09%	18:30	3.66%	19:00	2.26%
19:30	2.66%	20:00	3.19%	20:30	3.04%
21:00	4.30%	21:30	6.80%	22:00	6.80%
22:30	3.46%	23:00	5.31%	23:30	3.17%

Average gap = 2.615 Hrs.
Items used = 48

Figure 3.5 (continued)
Time Of Day Distribution Of Household Burglaries

Weekdays Autumn

00:00	1.73%	00:30	0.70%	01:00	1.50%
01:30	1.08%	02:00	3.25%	02:30	2.23%
03:00	1.66%	03:30	2.71%	04:00	1.40%
04:30	0.79%	05:00	0.69%	05:30	0.65%
06:00	0.71%	06:30	0.71%	07:00	0.50%
07:30	0.50%	08:00	0.40%	08:30	0.49%
09:00	0.70%	09:30	0.69%	10:00	1.55%
10:30	2.12%	11:00	1.80%	11:30	2.35%
12:00	1.90%	12:30	2.29%	13:00	2.38%
13:30	3.16%	14:00	3.99%	14:30	3.62%
15:00	4.28%	15:30	4.72%	16:00	3.14%
16:30	3.06%	17:00	2.04%	17:30	1.69%
18:00	2.13%	18:30	3.20%	19:00	3.23%
19:30	4.14%	20:00	4.06%	20:30	3.53%
21:00	2.88%	21:30	2.47%	22:00	3.47%
22:30	1.02%	23:00	1.44%	23:30	1.24%

Average gap = 2.497 Hrs.
Items used = 176

Weekends Autumn

00:00	1.94%	00:30	3.30%	01:00	3.71%
01:30	3.08%	02:00	2.38%	02:30	1.20%
03:00	1.00%	03:30	2.18%	04:00	3.31%
04:30	0.49%	05:00	0.35%	05:30	0.00%
06:00	0.47%	06:30	1.88%	07:00	0.47%
07:30	0.47%	08:00	0.47%	08:30	0.47%
09:00	0.28%	09:30	0.28%	10:00	0.42%
10:30	0.77%	11:00	0.90%	11:30	0.62%
12:00	0.62%	12:30	3.09%	13:00	2.00%
13:30	0.59%	14:00	1.53%	14:30	1.85%
15:00	1.49%	15:30	1.81%	16:00	2.30%
16:30	1.97%	17:00	2.92%	17:30	1.51%
18:00	3.05%	18:30	2.10%	19:00	2.50%
19:30	3.22%	20:00	5.45%	20:30	6.82%
21:00	4.84%	21:30	5.57%	22:00	3.91%
22:30	5.19%	23:00	2.49%	23:30	2.76%

Average gap = 2.296 Hrs.
Items used = 71

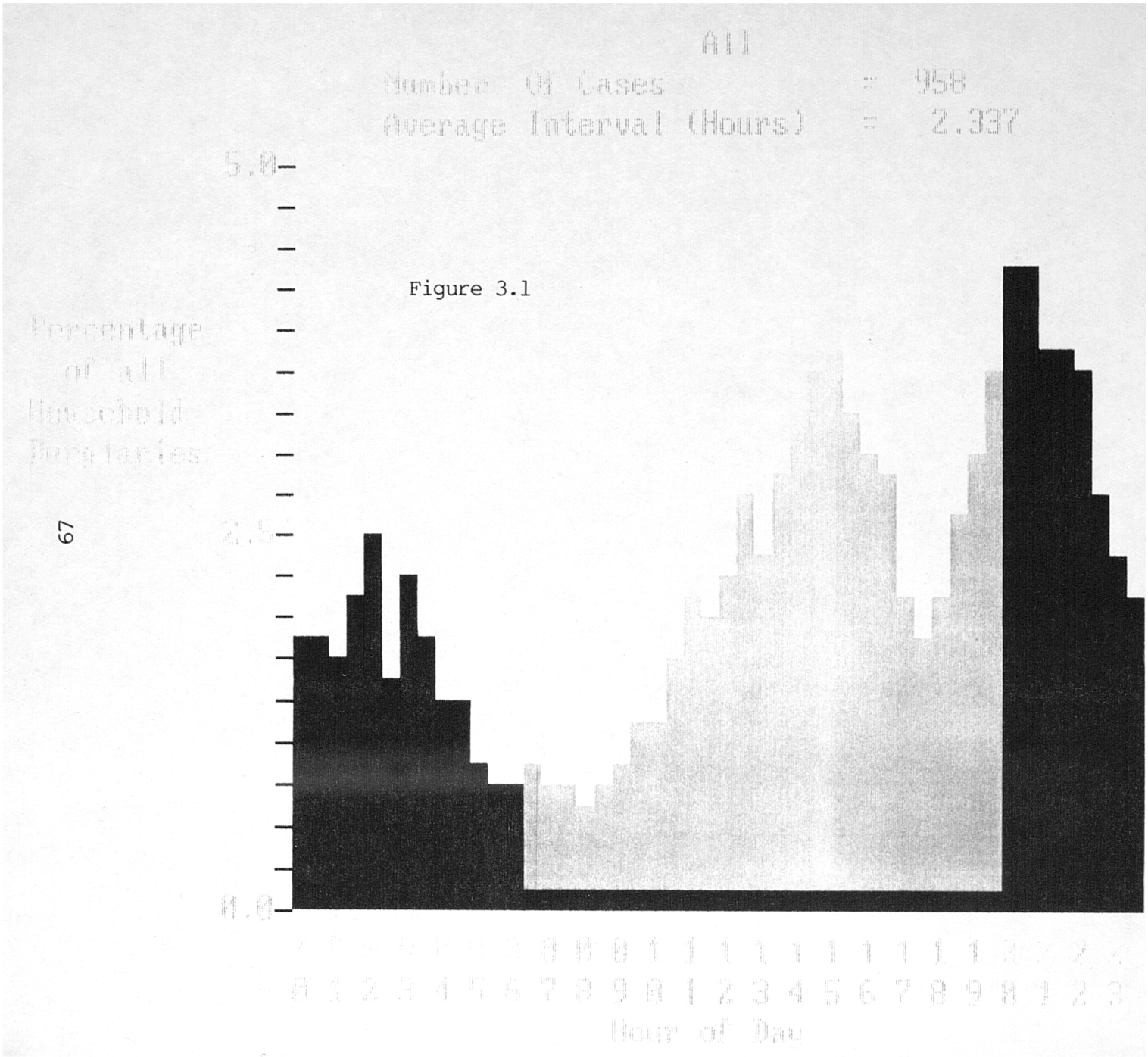

All
Number Of Cases = 958
Average Interval (Hours) = 2.337
5.0
2.5
0.0
Percentage of all Household Journeries
Hour of Day

Figure 3.1

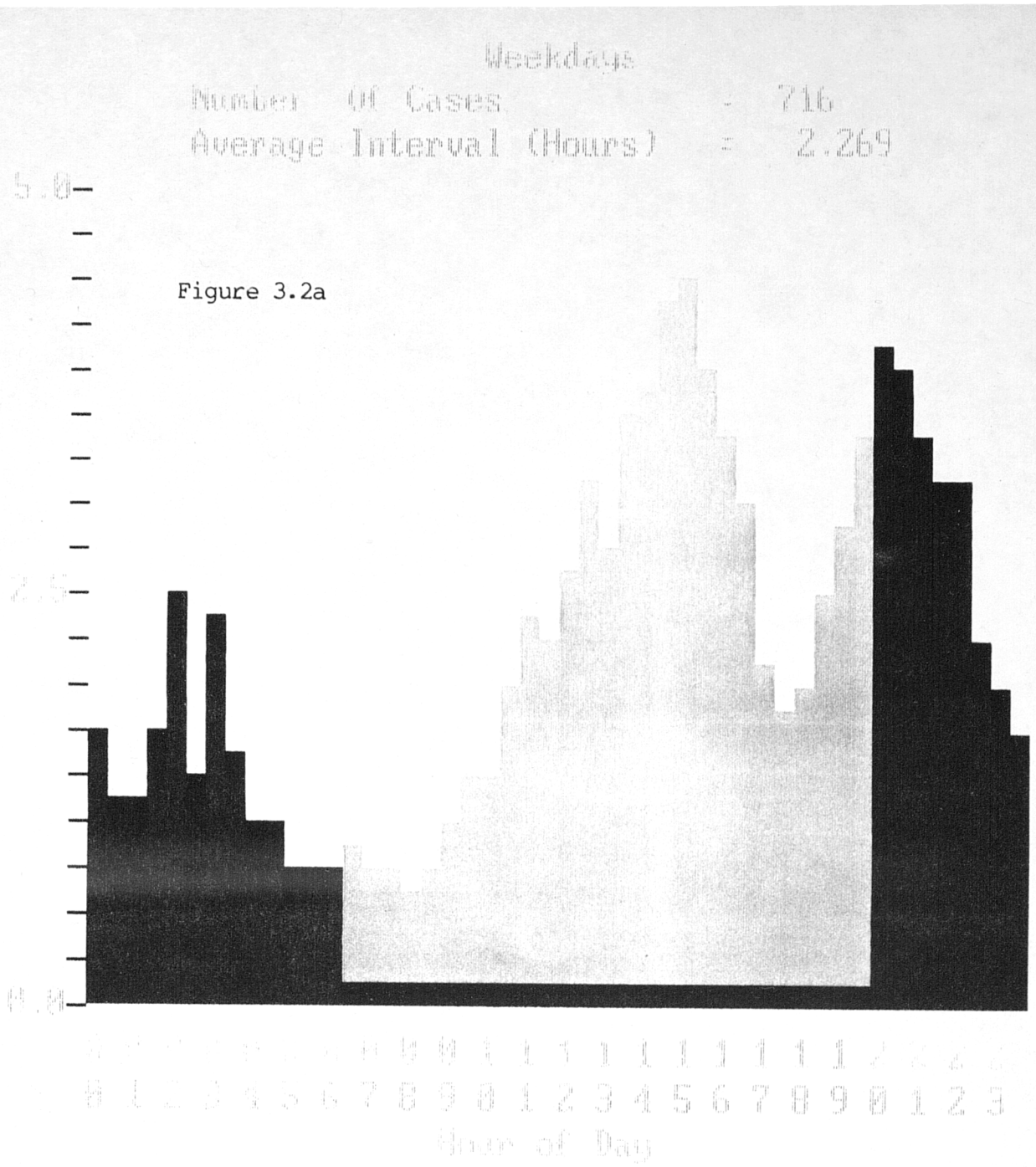

Weekdays
Number Of Cases = 716
Average Interval (Hours) = 2.269
5.0
Percentage of all Household Journeys
Hour of Day

Figure 3.2a

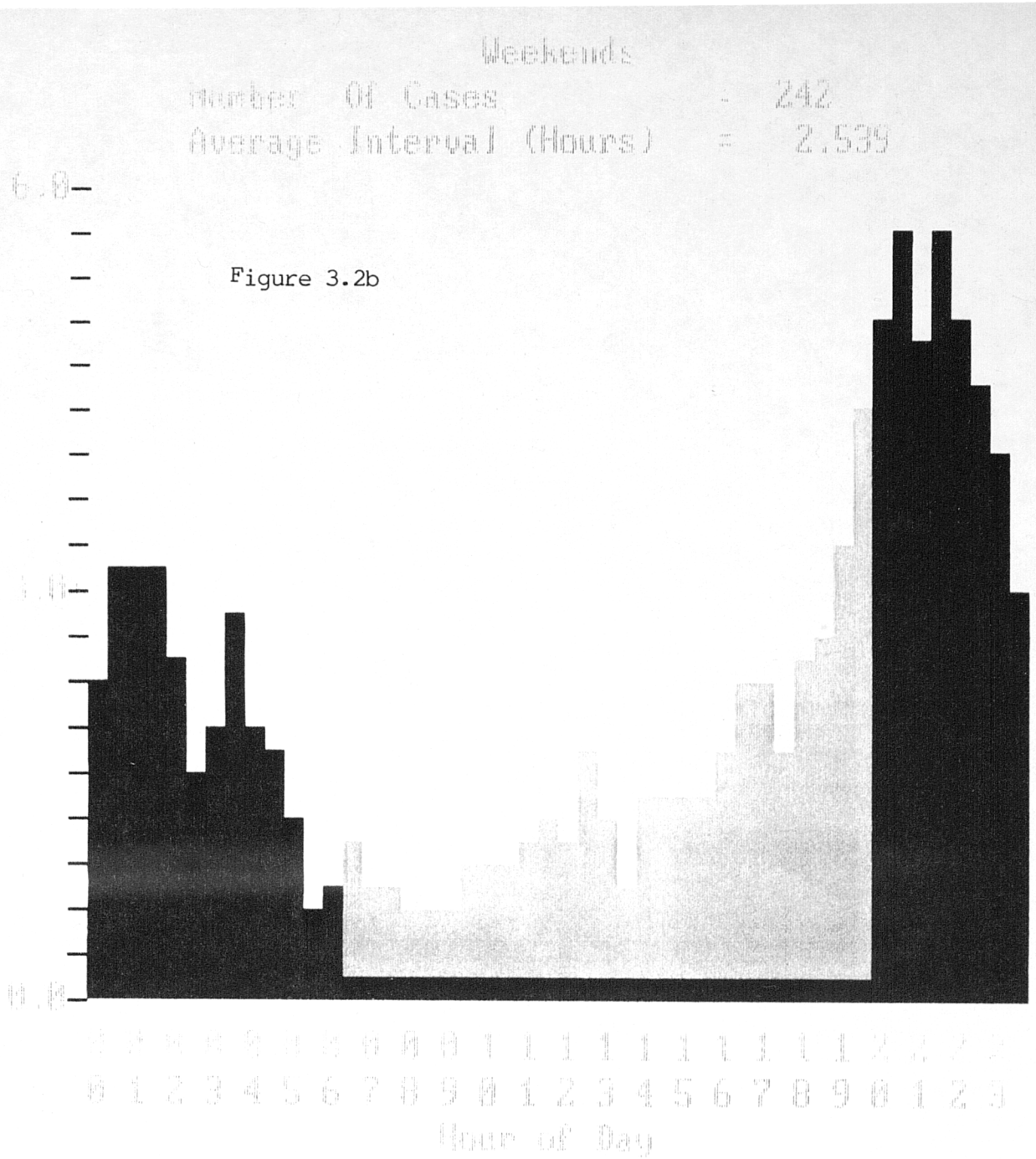

Weekends
Number Of Cases = 242
Average Interval (Hours) = 2.539
6.0
Percentage of all Household Registries
Hour of Day

Figure 3.2b

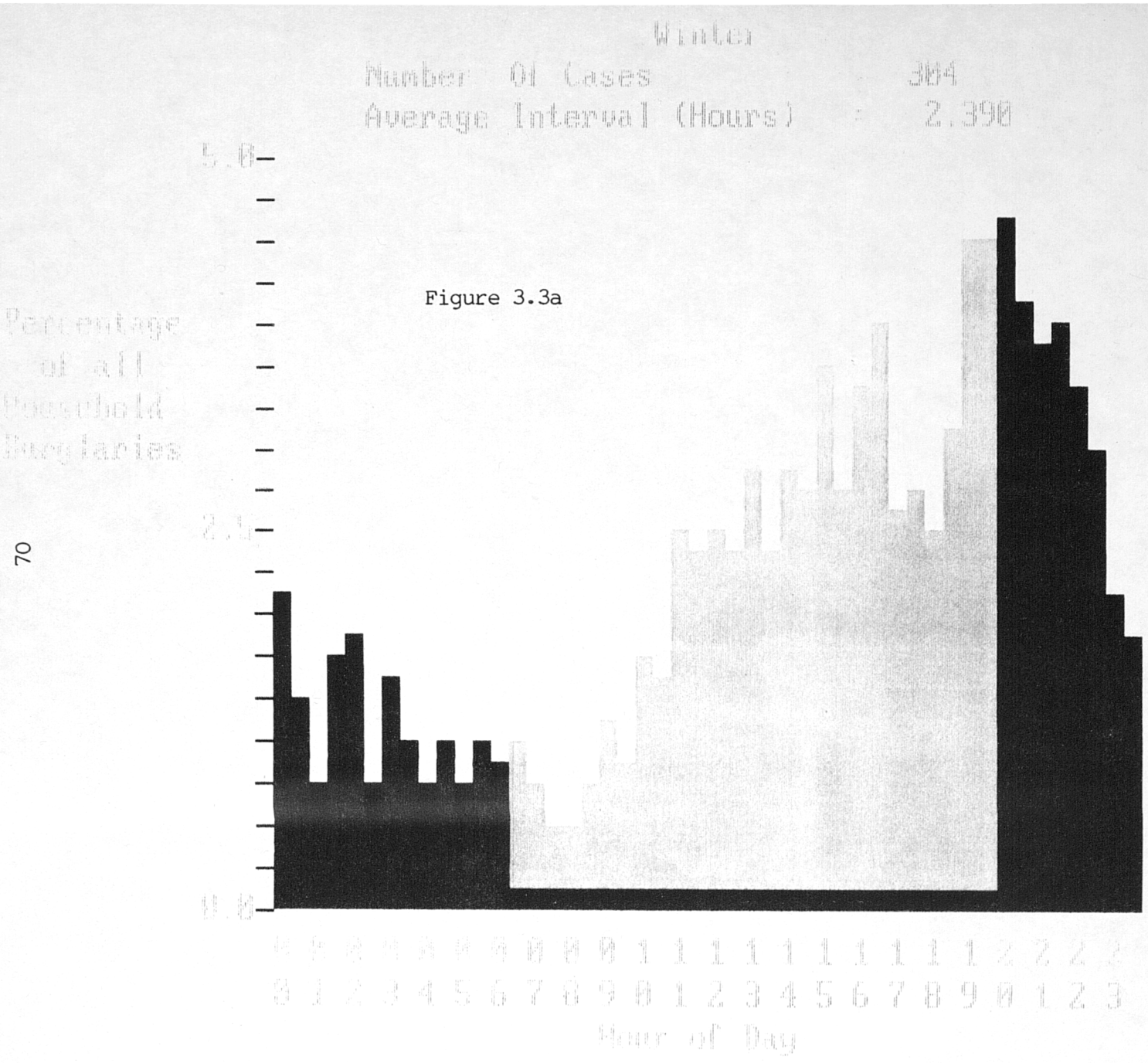

Winter
Number Of Cases : 304
Average Interval (Hours) : 2.390
5.0
2.5
0.0
Percentage of all Household Burglaries
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day

Figure 3.3a

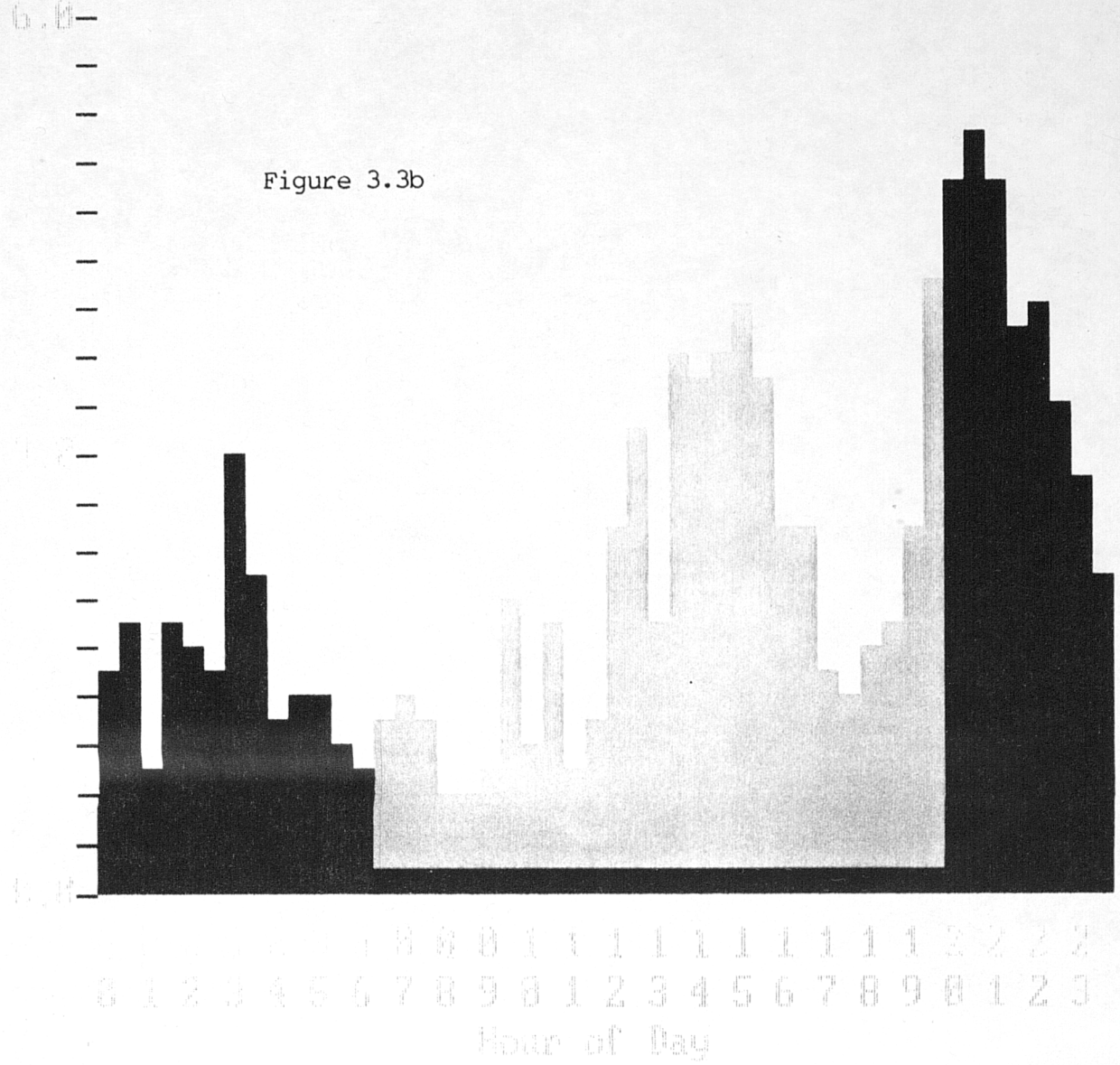

Spring
Number Of Cases : 202
Average Interval (Hours) = 2.307
6.0
Percentage of all Household
Hour of Day

Figure 3.3b

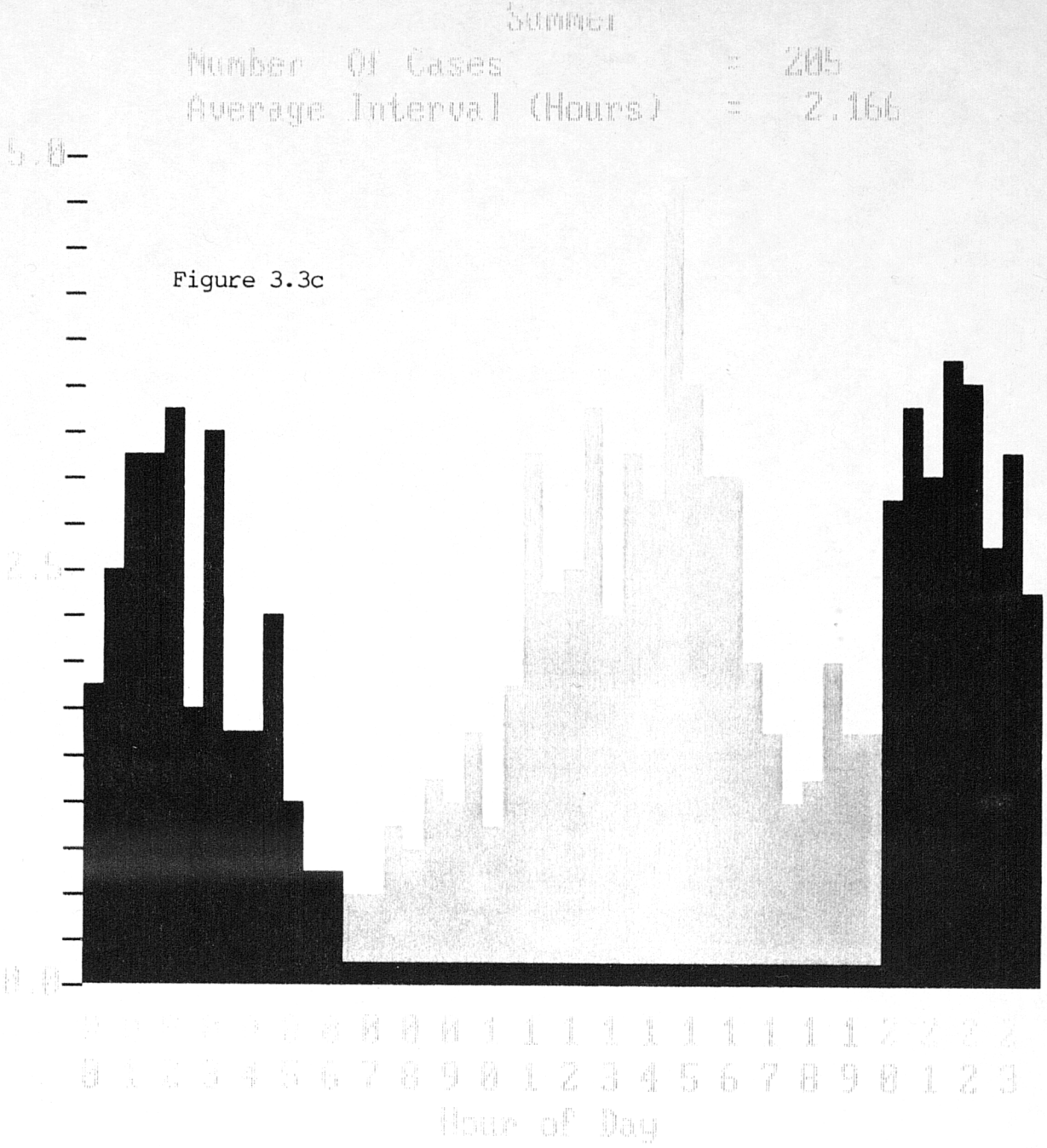

Summer
Number Of Cases = 205
Average Interval (Hours) = 2.166
5.0
Hour of Day

Figure 3.3c

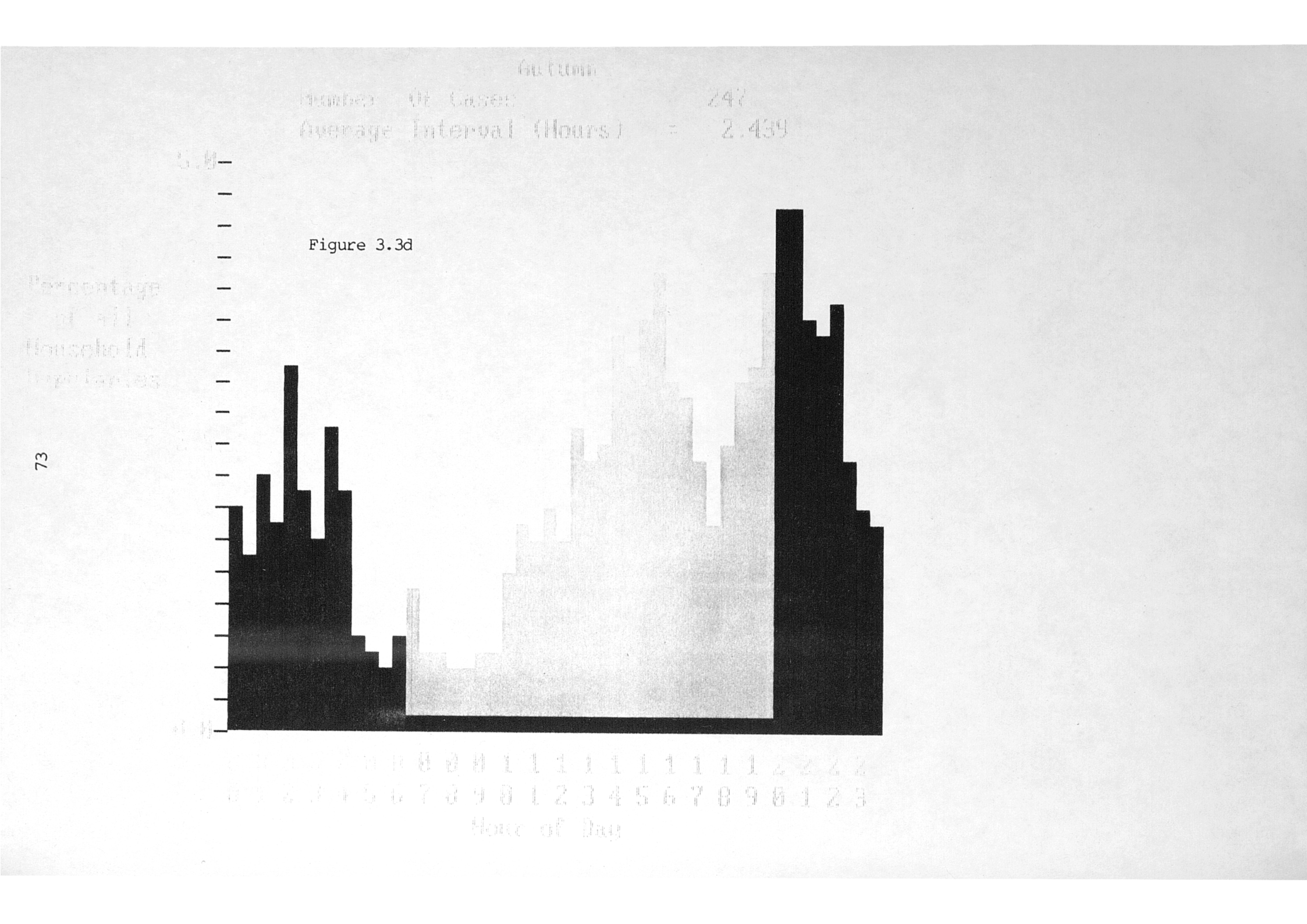

Autumn
Number Of Cases: 247
Average Interval (Hours) = 2.439
Percentage of All Household Departures
Hour of Day

Figure 3.3d

Figure 3.4a

Figure 3.4b

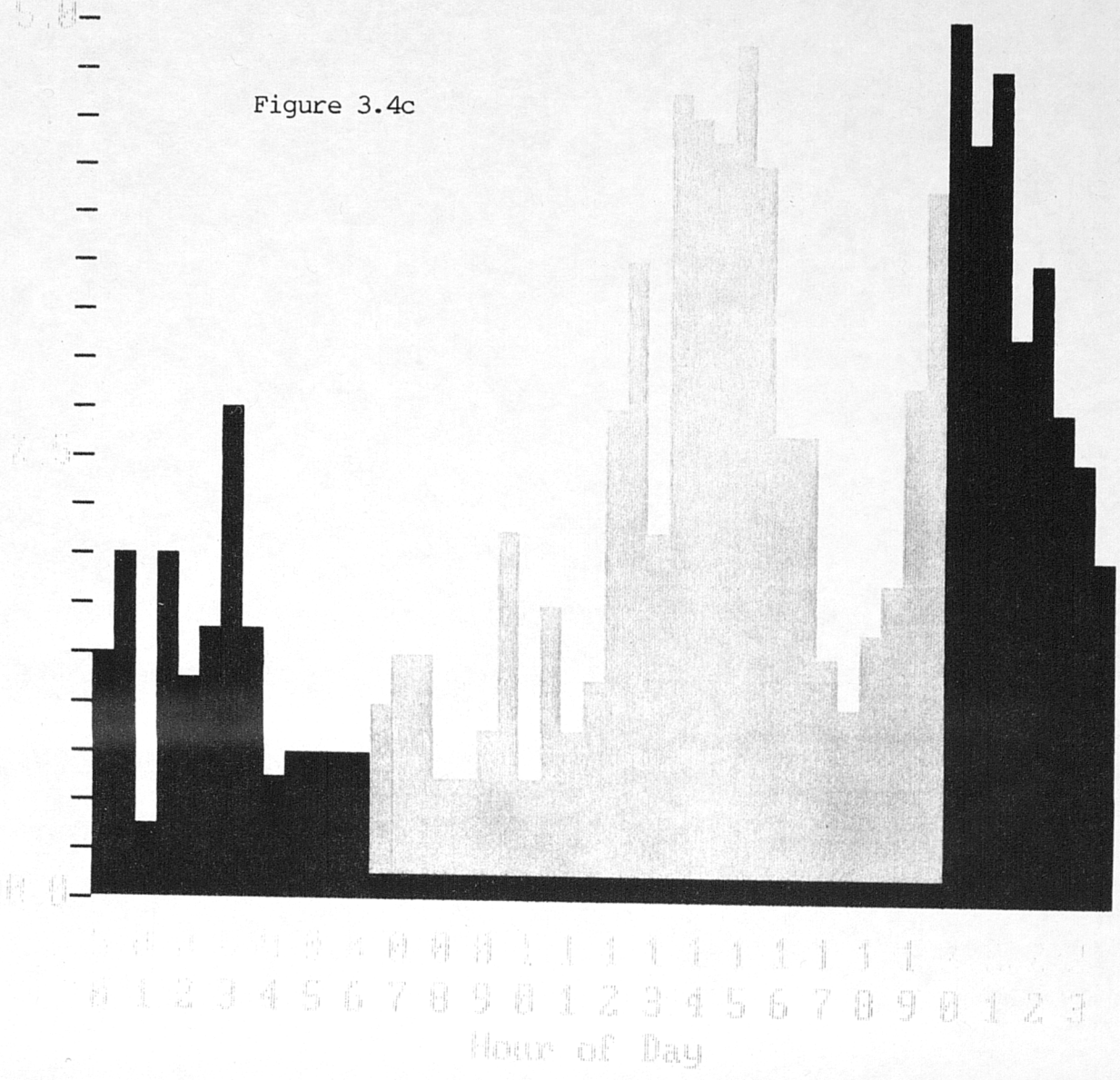
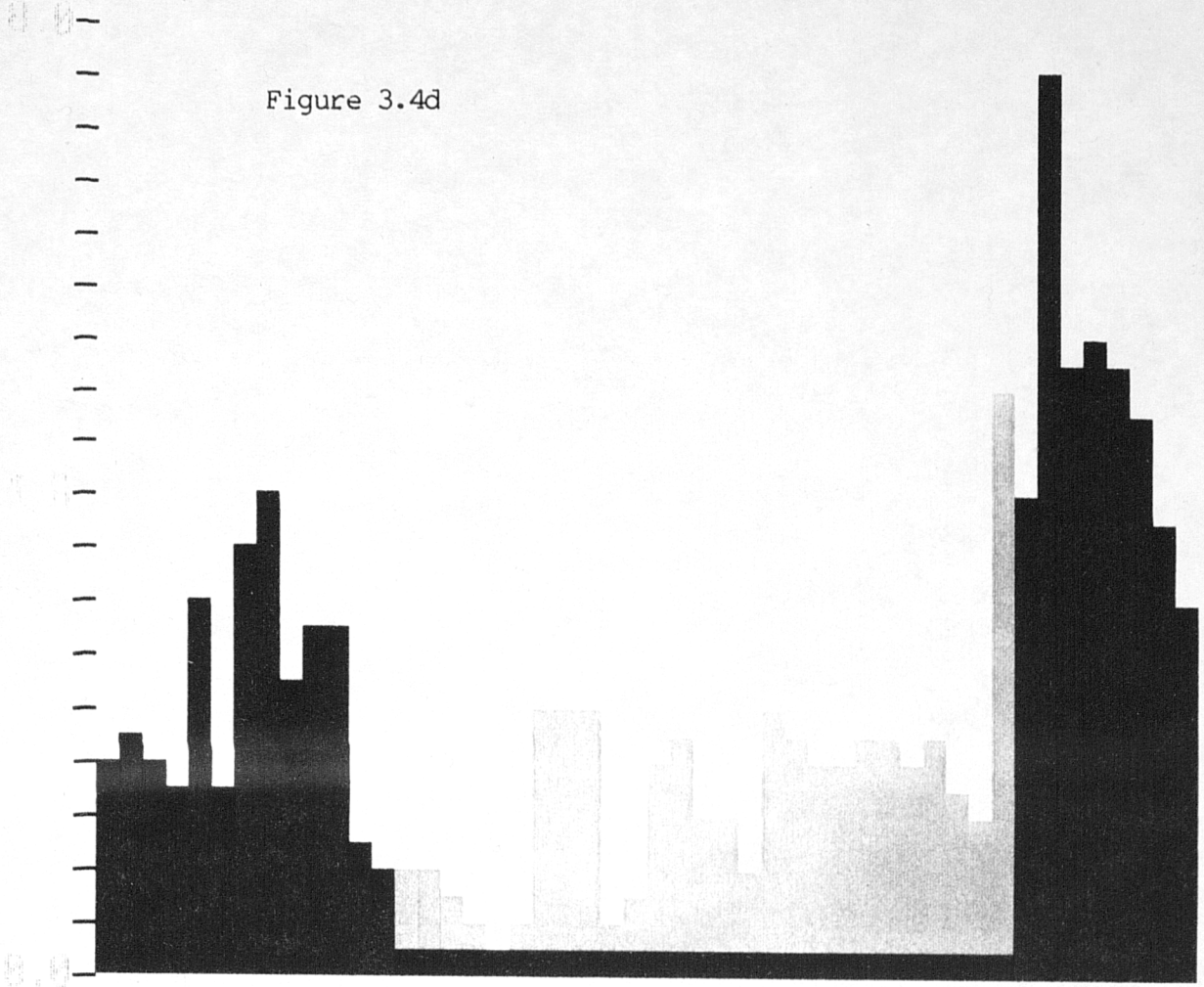

Weekdays Spring
Number Of Cases = 147
Average Interval (Hours) = 2.214
Hour of Day

Figure 3.4c

Figure 3.4d


Weekends Spring
Number Of Cases = 55
Average Interval (Hours) = 2.555
Hour of Day

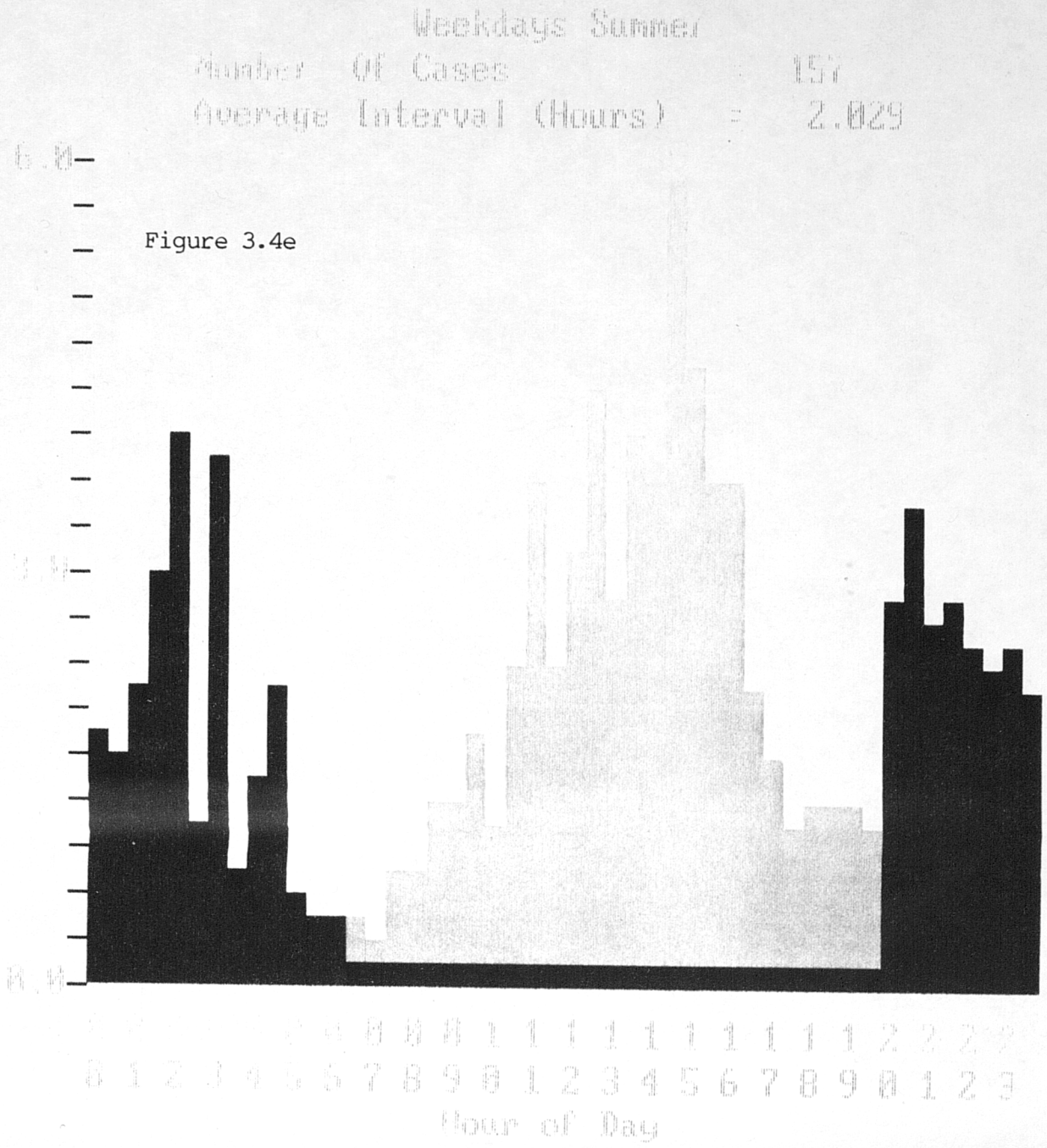

Weekdays Summer
Number Of Cases : 157
Average Interval (Hours) = 2.029
6.0
Percentage of All Households Boardings
Hour of Day

Figure 3.4e

Figure 3.4f

Figure 3.4g

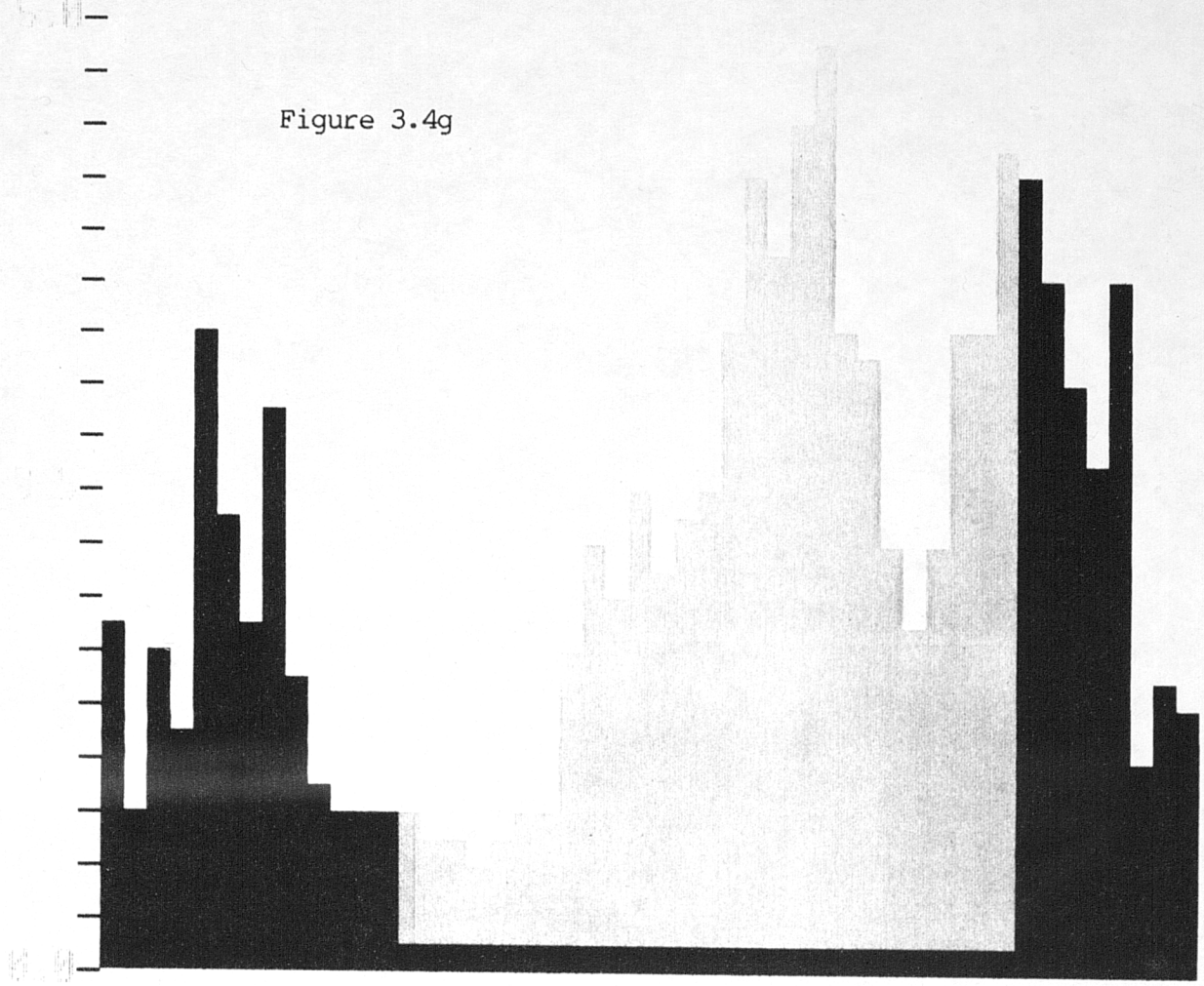

Weekdays Autumn
Number of Cases : 176
Average Interval (Hours) : 2.497
Hour of Day

Figure 3.4h

times of day appear to be between 7.00 and 9.00 hours, and again between 17.00 and about 19.00 hours. These times seem to correspond to times when households are most likely to be at home. This suggests that likelihood of entering a house when some of its occupiers are present may be some form of deterrent to burglars.

This idea is given further support from the weekends-only data. Here, the peak times are similar to the weekday times, except there is no peak risk in the afternoon. Clearly a typical working household is more likely to be in on weekend afternoons than on weekdays. It is certainly possible that whether the householder actually is more likely to be in during the day at weekends or not, the burglar perceives the risk of being seen to be higher at these times.

The seasonal effects are perhaps less marked, although the afternoon peak appears later in winter and autumn than in spring or summer. This may perhaps be explained by the extra cover that darkness provides in those seasons, is the later part of the afternoon.

When the doubly split data sets are examined, allowing for the difficulties of small sample size, the results seem to suggest an overlaying of the two individual effects, rather than any interactive phenomena, where a particular pair of circumstances totally alters the pattern.

A final problem must be considered concerning the filtering out of observations with intervals exceeding a given time limit. It has already

been observed that large intervals tend to "smooth" the risk profile, and alternate the effects of peaks and troughs. It is possible, however, that the expected lengths of these intervals may not be uniform throughout the twenty-four hour period. The effect of this may be to reduce the "peakedness" of some high and low risk points. In particular, the early morning (2.00) peak may be reduced in effect, as occupiers may only discover burglaries on rising, without being able to specify the point during the night at which they occurred.

An associated effect is that by filtering out intervals over a certain length, when length is non-uniform on average throughout the day, more information will be lost at certain times of day than on others. The study would not be complete without considering the effect of this on the current dataset.

This will be done using time methods. Firstly, the average interval length will be computed, with the day being split into three 8-hour periods. After this, the effect on the overall shape of the risk profile will be considered when the maximum interval cut-off time is considered.

The first method will now be considered. Clearly, exact time of incident is not known, so that this will be estimated as mid-interval here, and then classified into one of three categories, (midnight to 8am, 8am to 4pm, 4pm to midnight) and average length of these considered. The results are given in table 3.6.

Table 3.6
Length Of Uncertainty Interval By Time Of Day

Time Of Day	Average Length
Midnight to 0800	6.74
0800 to 1600	5.67
1600 to Midnight	6.24

In fact, there is not a great deal of variation between the average interval length for the three day-time categories. This suggests that, although greatly varying interval lengths throughout the day may cause problems, this does not happen here.

The second technique, of interval cut off variation is illustrated in figure 3.5. Four cut-off points are shown, at 4, 6, 8 and 10 hours. As the results of the last test might imply, again there is little difference between the four graphs, except that when the cut-off point is made more restrictive, the sample is reduced and "spikeyness" of the graph is emphasised. Thus, it seems reasonable to conclude that the original data analysis, with its associated risk profiles may be considered accurate, without the "smoothing" distortion mentioned earlier.

3.2.5 Discussion

The results of this study may be considered in two contexts: firstly as a study in its own right, and secondly from the viewpoint of the PhD as a whole. As a stand alone study, some of the results are interesting. They seem to support two ideas often put forward by police officers, and criminologists. Firstly, that burglaries mostly happen to houses when they are empty; the times of day when burglary least occurs appears to be when occupiers are most likely to be in: breakfast and supper times during the week and daytime at weekends. The second idea supported is that of "defensible space": in autumn and winter, afternoon burglaries peak later on, when it is dark, and parts of neighbourhood generally visible to residents in the daytime are no longer

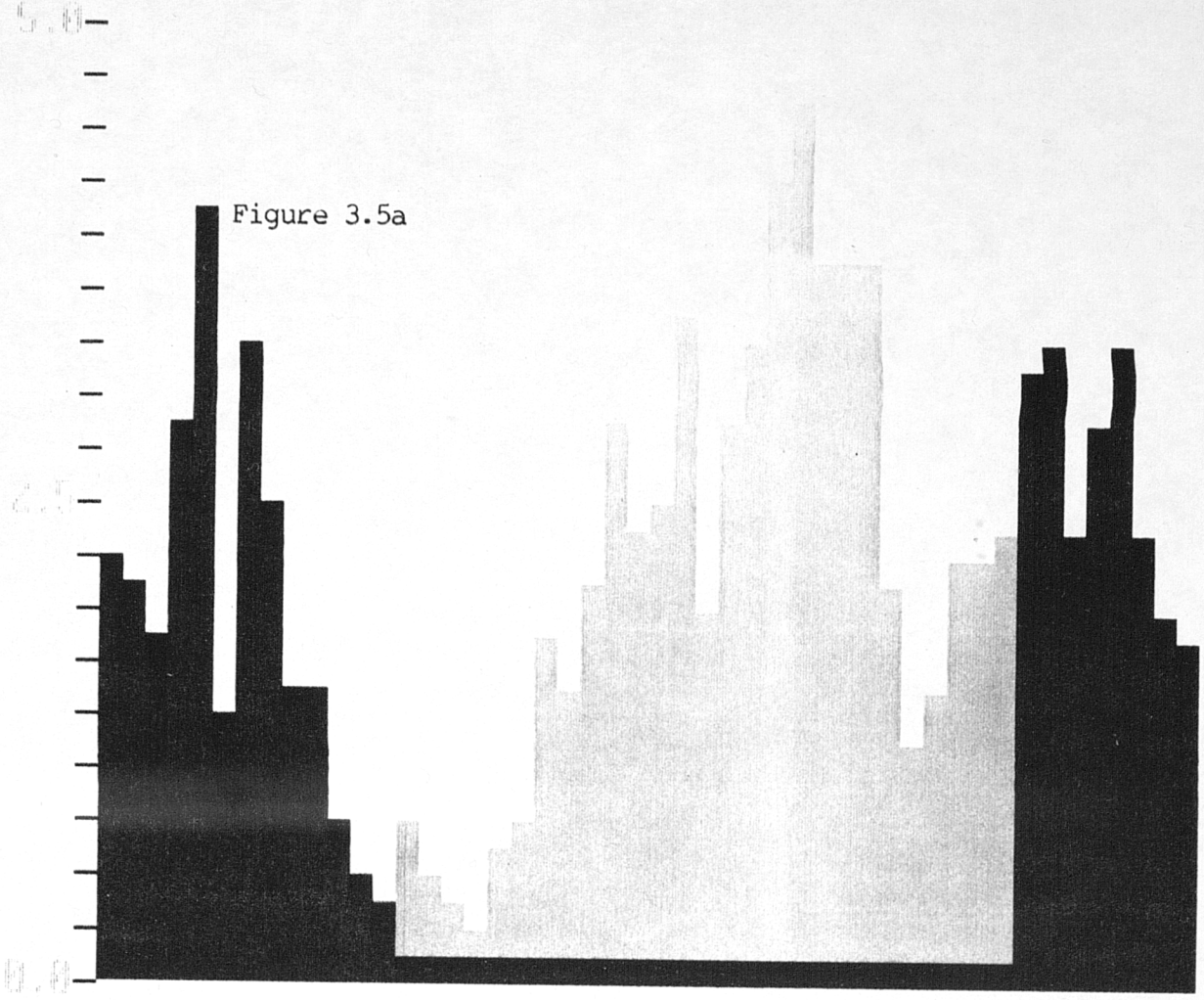

All
Number Of Cases = 396
Average Interval (Hours) = 0.511
5.0
2.5
0.0
Percentage of All Household Burglaries
Hour of Day

Figure 3.5a

Figure 3.5b

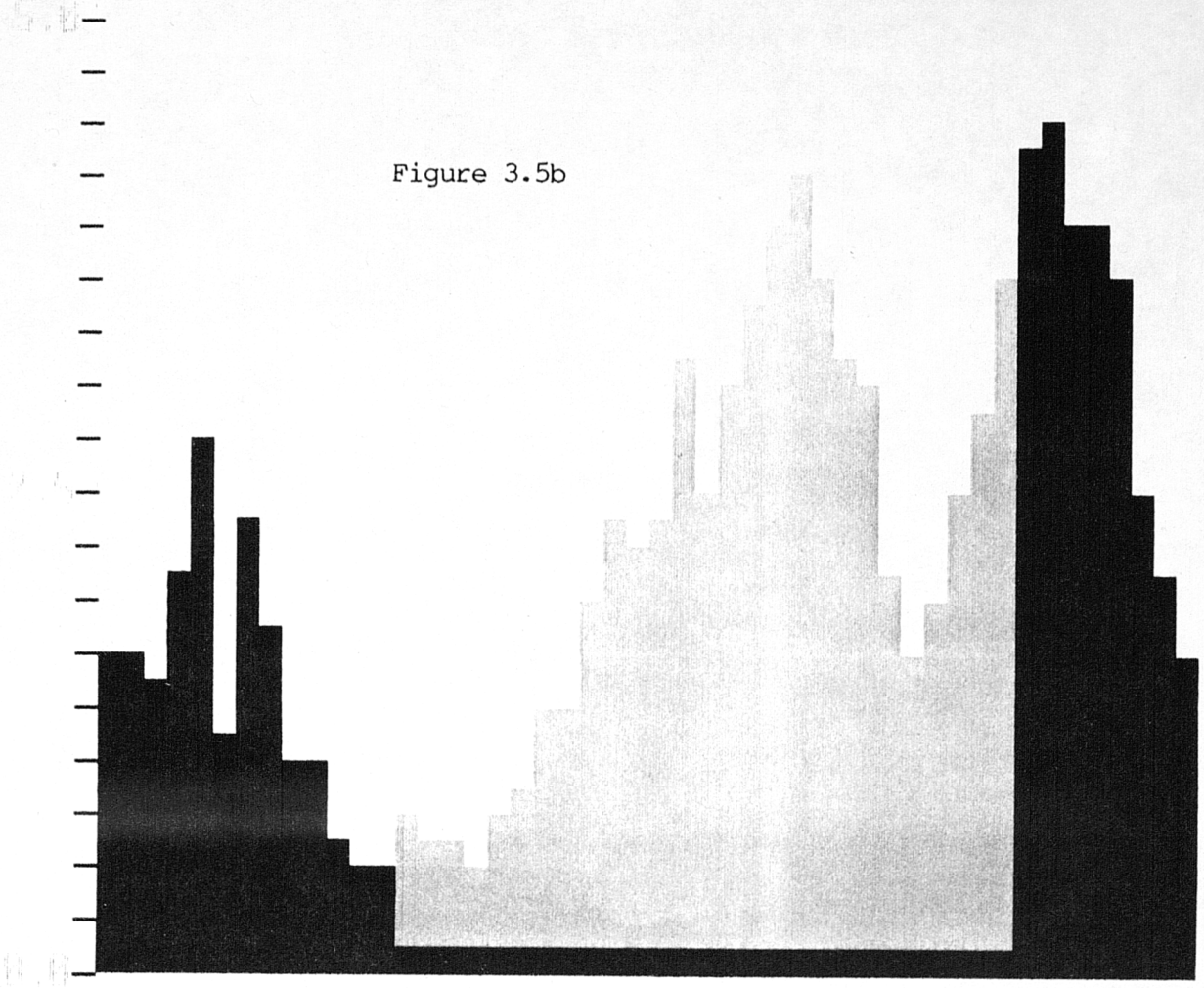


Figure 3.5c

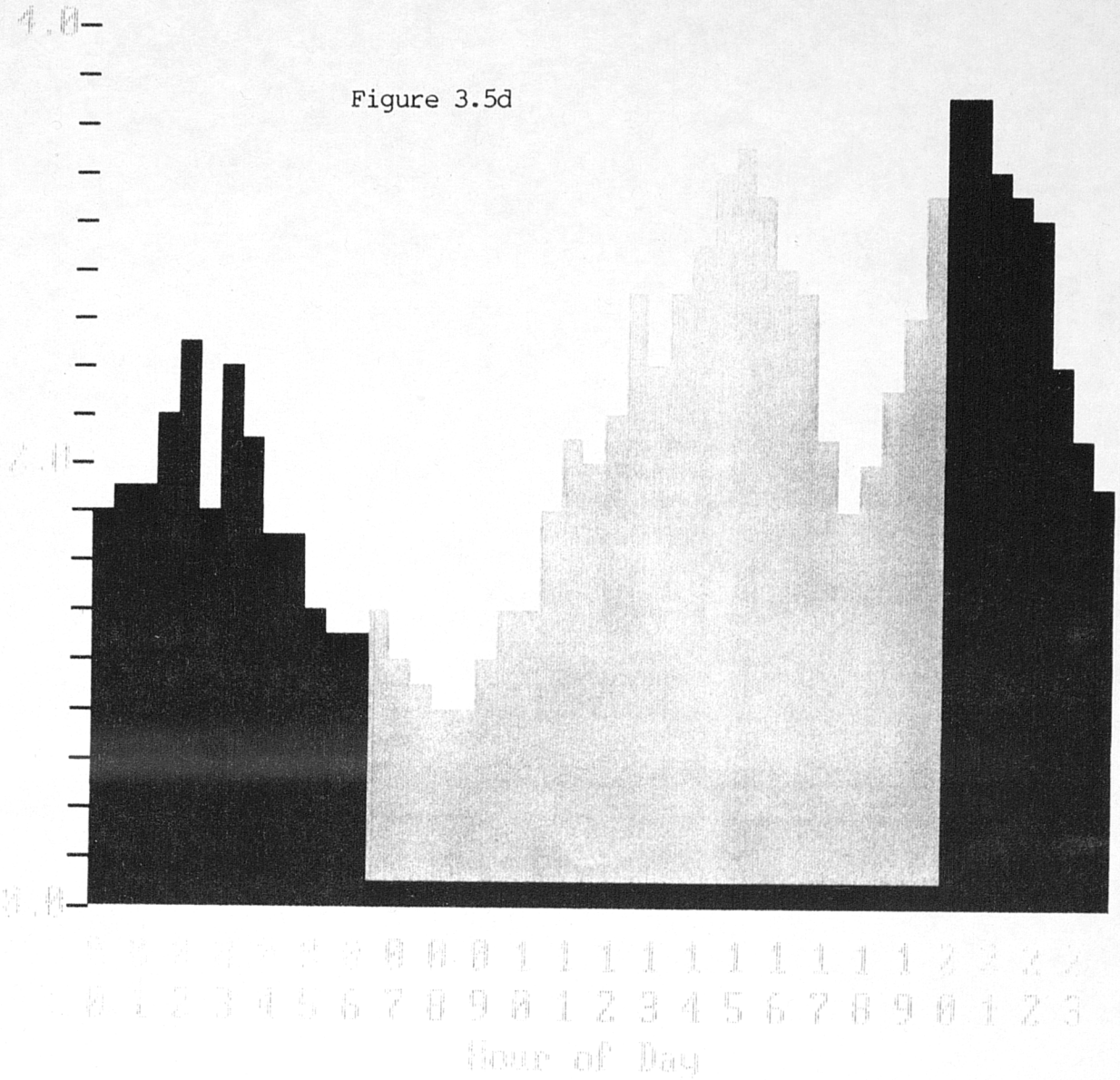

All
Number Of Cases = 1182
Average Interval (Hours) = 3.170
4.0
Percentage of All Household Burglaries
Hour of Day

Figure 3.5d

exposed. In general policing terms, these results are also useful. Clearly it is of benefit for foot beat officers (and car patrols) to be aware of high risk times for household burglary, when on street patrol.

In the wider context of the overall crime pattern analysis system, however, attention has to be given to the generally wide nature of the intervals. In the case of the data analysis here, if there is an interval exceeding 6 hours, the item of data is discarded. This results in losing 50% of the sample. However, in the data were also subdivided by foot beats, this would result in very low item counts in each category. Thus, it is unlikely that analysis of this sort could be applied on a beat-by-beat basis. It also seems unlikely that it could be used in a predictive model, again due to the large degree of uncertainty of exact time of burglary in much of the data.

It may be possible, however, to use the program developed here as part of an overall crime pattern analysis system, running on a separate, subdivision wide database. Thus, although unable to identify intra-beat variation in risks during the day a general impression of risk throughout the area may be gained. Also, the software could, at some later stage, be extended.

For example, in addition to recording the time interval during which the burglary was thought to occur, the time that the event was reported is also stored. This allows Kernel estimators of crime report frequencies to be built up, in addition to the frequencies of the crimes themselves.

This may have implications for responsive, rather than preventative policing.

The relationship between time of occurrence, and time of reporting of crimes might also be examined. For example, several crimes occurring at different times in the night might all be reported at a similar time in the morning, when they are discovered. In the daytime, when householders are out for shorter periods, however, reporting may be more evenly spread.

Thus, although the analytical techniques here may not by reasonably incorporated into a subdivisional software system in a geographical context, they may justify inclusion in their own right, for the management of the sub-division as a whole. At divisional level, of course, intra-subdivisional geographical variations could be analysed, which may yield some useful information for resource management at a larger scale of resolution.

3.3 Examination of Seasonal Variation in Household Counts

In addition to geographical factors affecting rates of crime within a subdivisional level, some variation may also be accounted for by season effects. For example, increase in the hours of darkness may bring about increased risk of household burglaries. Alternatively, some crime may depend on the weather - potential burglars may decide against activity in adverse conditions. As well as general effects such as these, there are some more specific seasonal phenomena leading to increased

likelihood of crime. Certain public holidays occur regularly on a yearly basis, such as Christmas and Easter, during which people may customarily leave their homes for a number of days. This may leave the houses vulnerable to household burglary.

Thus, several arguments suggest that there should be a certain amount of seasonal pattern in crime rates. It is also possible that these seasonal patterns may vary geographically. Weather conditions and hours of darkness vary over space, and although certain public holidays may be fixed nationally, other local events may exist, bringing householders away from their houses.

From this, it may be argued firstly that if regular seasonal patterns are discovered in crime counts, then presenting this information to Police officers would help in short term planning of resource management, and also aid the prediction of crime in the near future. Secondly, it may also be argued that, since these patterns are liable to vary geographically, any seasonal variation analysis presented to officers working in a given subdivision should be based on data from that subdivision, and not, for example, from National or even Force-wide data.

The intention of such an analysis would be mainly descriptive. If weekly crime counts are compiled, a weekly seasonal average may be computed, and plotted on a computer display, for example. Police officers with a knowledge of the locality may then be able to identify causes for particular effects. However, even if regular patterns occur

without explanation the identification of the patterns will be of use, in a predictive sense. It is also possible that identification of certain seasonal effects may lead to investigation and insight into certain crime patterns that had previously gone unnoticed.

In this exploratory analysis, then, it is proposed to examine the household burglary data discussed in Chapter 2, to discover any seasonal patterns that may exist. The exploratory examination may also yield some results that may be carried through into the crime prediction method that is one of the principal aims of this PhD. Also, in order to analyse this data, some discussion arises as to how results may be presented. Both the means of presentation, and the results of the analysis are to be considered here.

3.3.1 Presentation of Data

There are two main uses for this type of data: firstly to identify any regular patterns that occur from year to year, and secondly to compare a given year with the previous year. In the first case, regularly occurring peaks or troughs are to be identified. In the second case, at any given week the previous years cumulative total up to that week is to be compared with the current year. This can be thought of as using the past years cumulative crime counts as "target" levels and to attempt to keep counts for the current year below these levels. Since figures here are cumulative, after a particularly bad month, a response in the following month may bring figures back down to the target. however,

failing to compensate would leave cumulative figures still above this target.

In both the cases of the cumulative and non-cumulative analyses, it is clear that some visual indication of how weekly rates compare with their neighbours in time is important. In the cumulative case, attempts to keep crime rates down from the previous year need to be examined over time, and in the non-cumulative instance, peaks and troughs of incidence require measurement. This suggests that data needs to be represented in graphical, rather than tabular form. For each case, the needs of graphical representation will now be considered in turn.

Firstly, consider the simple seasonal rate graph. This could take the form of a bar graph, with a single bar for each week over a 52 week period. However, as it is important to compare the seasonal patterns over a period of several years, some means of rapidly switching between yearly bar graphs is required, or some means of overlaying. For printed output this is relatively simple; on a VDU reasonably fast refresh of a screen display is necessary, (or a facility to "overlay" new data on old).

For the cumulative analysis, two years rates are overlaid. It is important to demonstrate, in a visual display, whether this years results exceed the previous years, or are exceeded by then. Colour-coding of the display seems a reasonable means of doing this. The data will be displayed in the form of cumulative bar graph when the current years total exceeds the previous years, the discrepancy will be shaded red; if

the converse holds, the shading will be in blue. A computer program to display this data is given in listing 3.2, making use of the graphics characters available on an IBM PC in text mode.

3.3.2 Analysis of Data

The three-year beat-by-week matrix of household burglary data for 1984-86, as described in chapter 2 was used as a trial dataset for the seasonal analysis. Thermal wax copies of the output screens obtained when running the programs are given in figure 3.6. Firstly, consider the non-cumulative data. For the second and third years, the pattern is fairly similar. A certain amount of week-to-week variation occurs, and their seems to be a lower average level of weekly crimes in the mid part of the year. In absolute terms the number of crimes occurring seems not to alter greatly, being around fifty in the winter, and slightly lower in summer. In contrast to this, the first year's pattern is somewhat different. Although there is a drop over the later summer months, as experienced in the other years, there is an extreme increase in crime between April and May. The household burglary counts for the subdivision here are over 100, altogether larger than for any counts in the following two years.

Examining the cumulative curves reflects this discrepancy of the first year of study. For most of the second year, the cumulative total falls well below that of the first, due to the two "spikes" in April and

Figure 3.6a

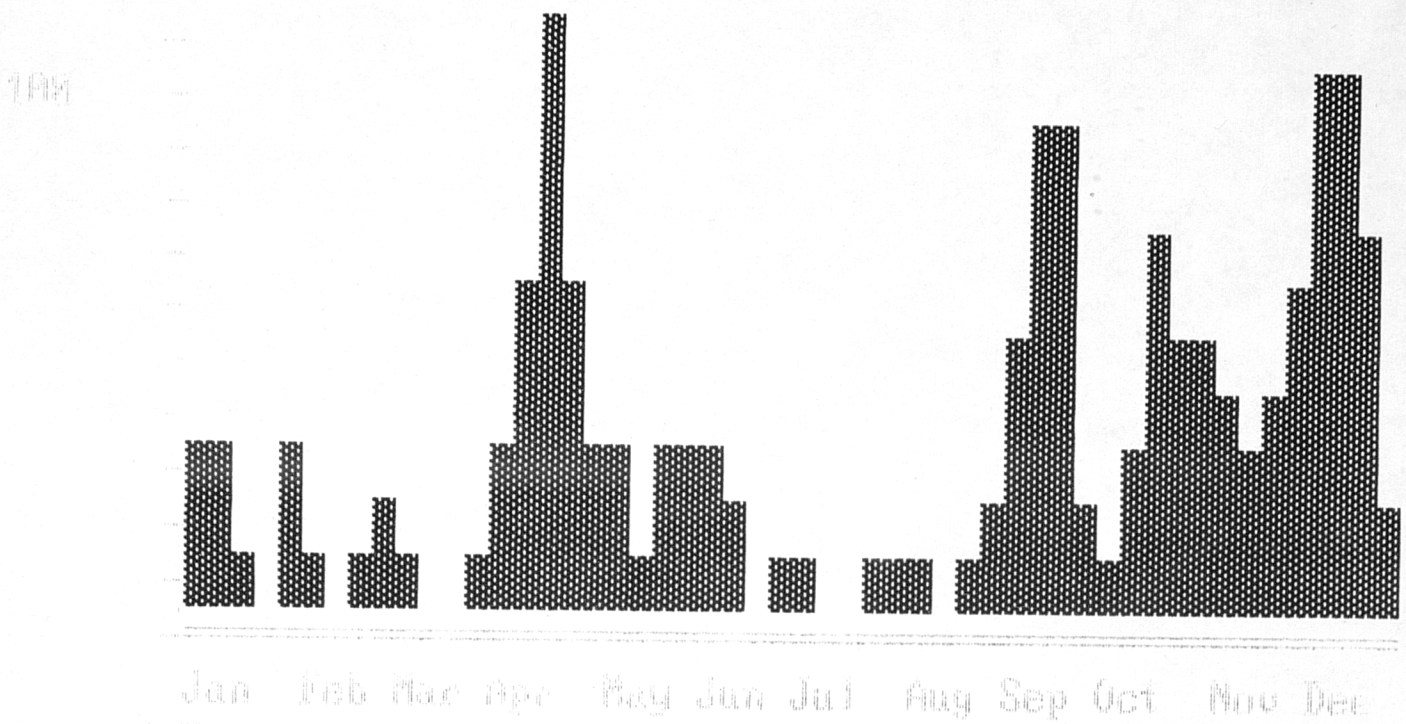

Seasonal Household Burglary Variation
Weekly Counts
200
100
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
For Other Options 1 - Next Year 2 - Quit

Figure 3.6b

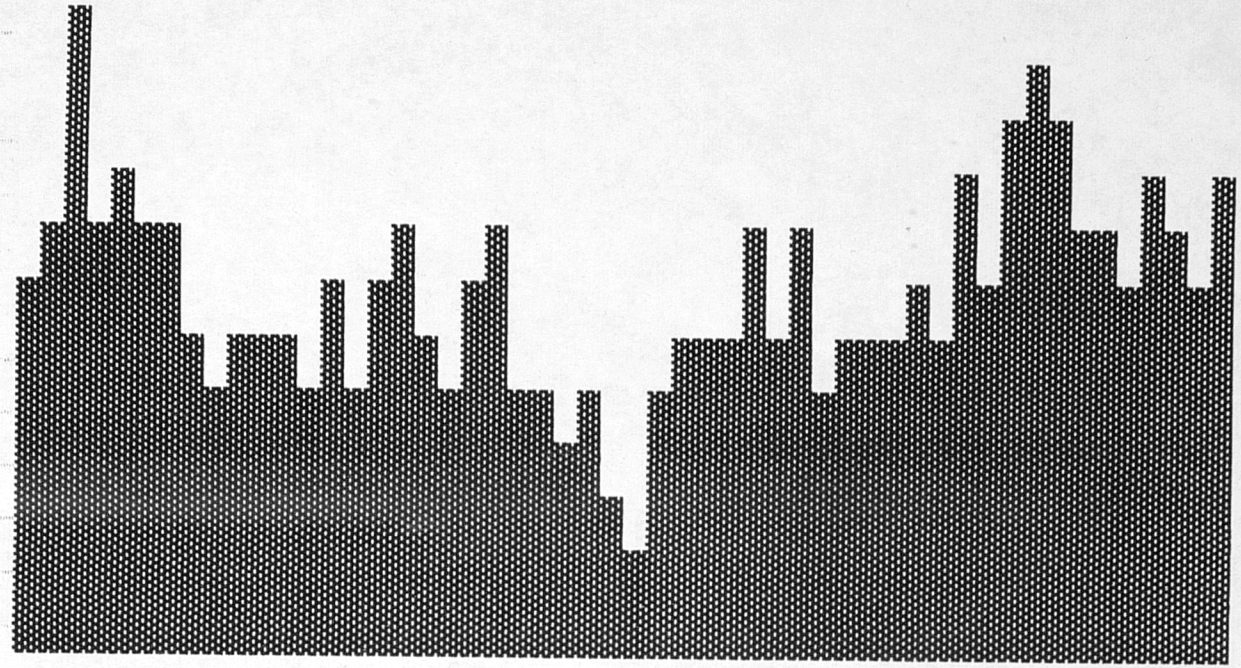
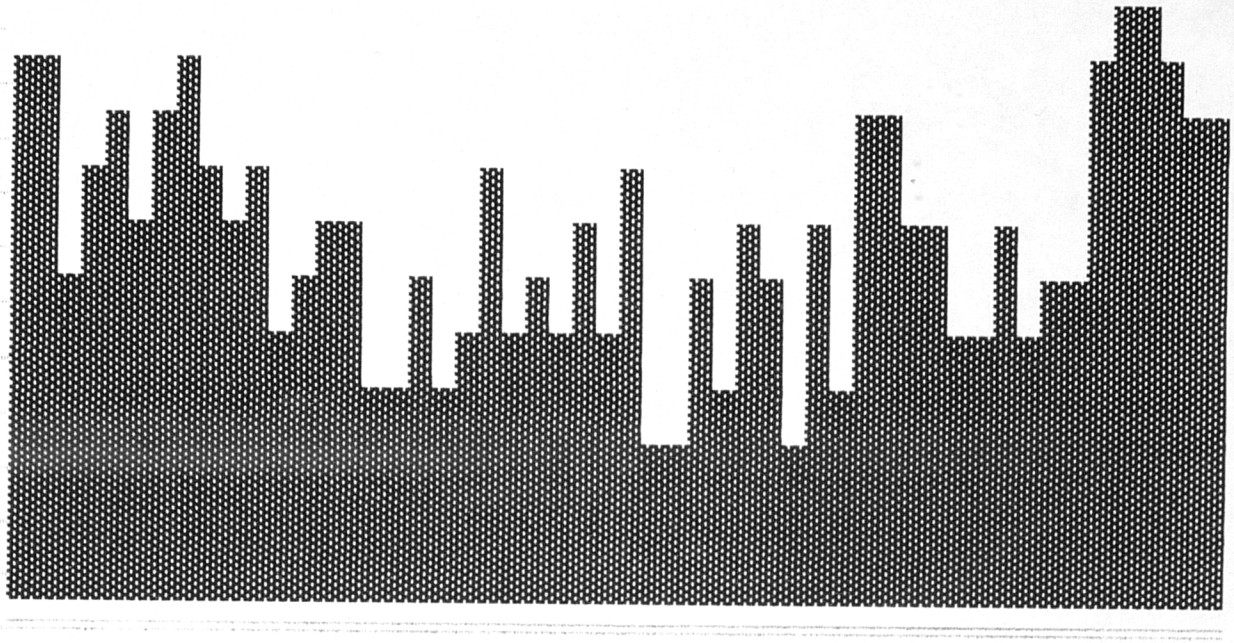

Seasonal Household Burglary Variation
Weekly Counts
100
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Further Options 1 = Next Year 2 = Quit

Figure 3.6c


Seasonal Household Burglary Variation
Weekly Counts
100
50
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Further Options: 1 = Next Year 2 = Quit

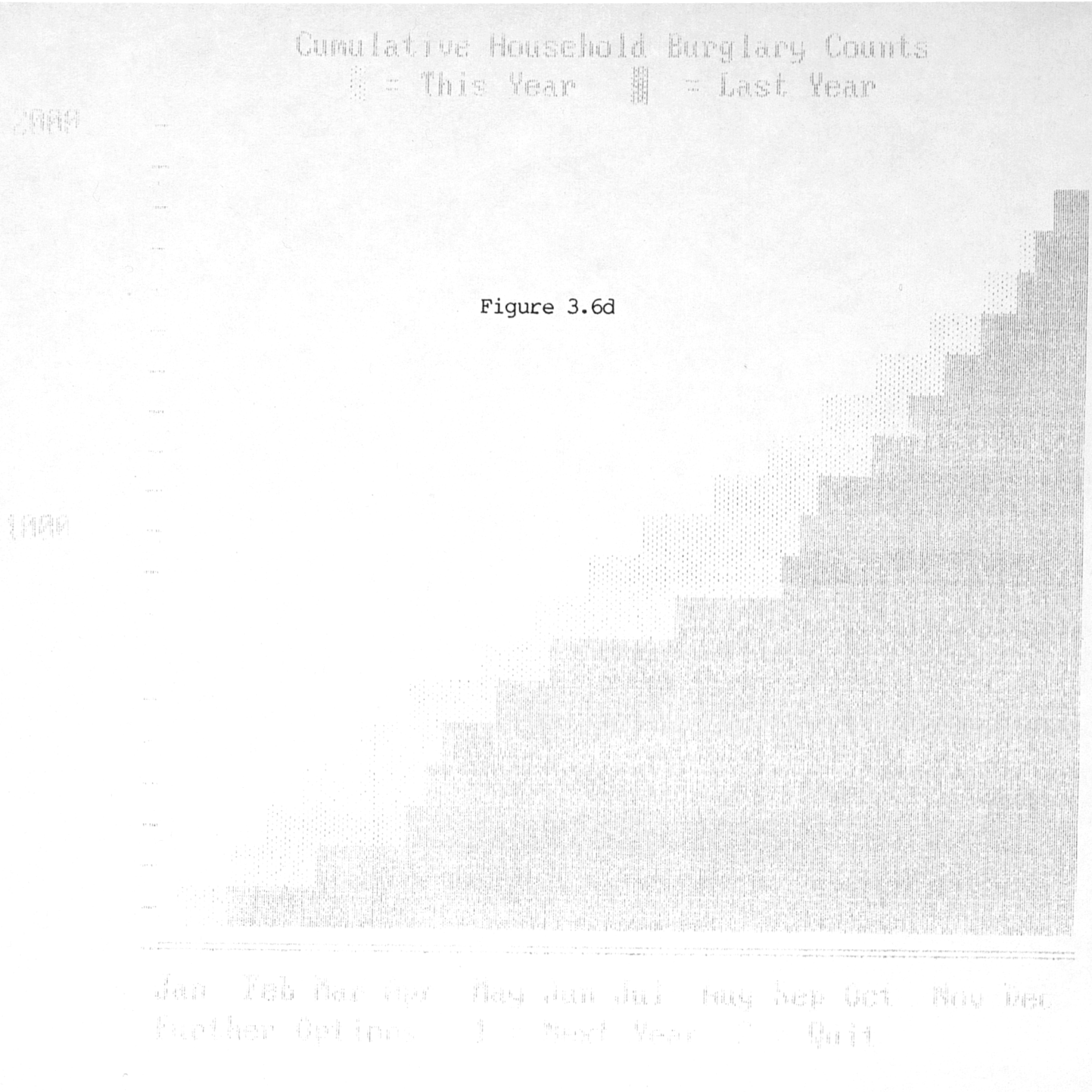
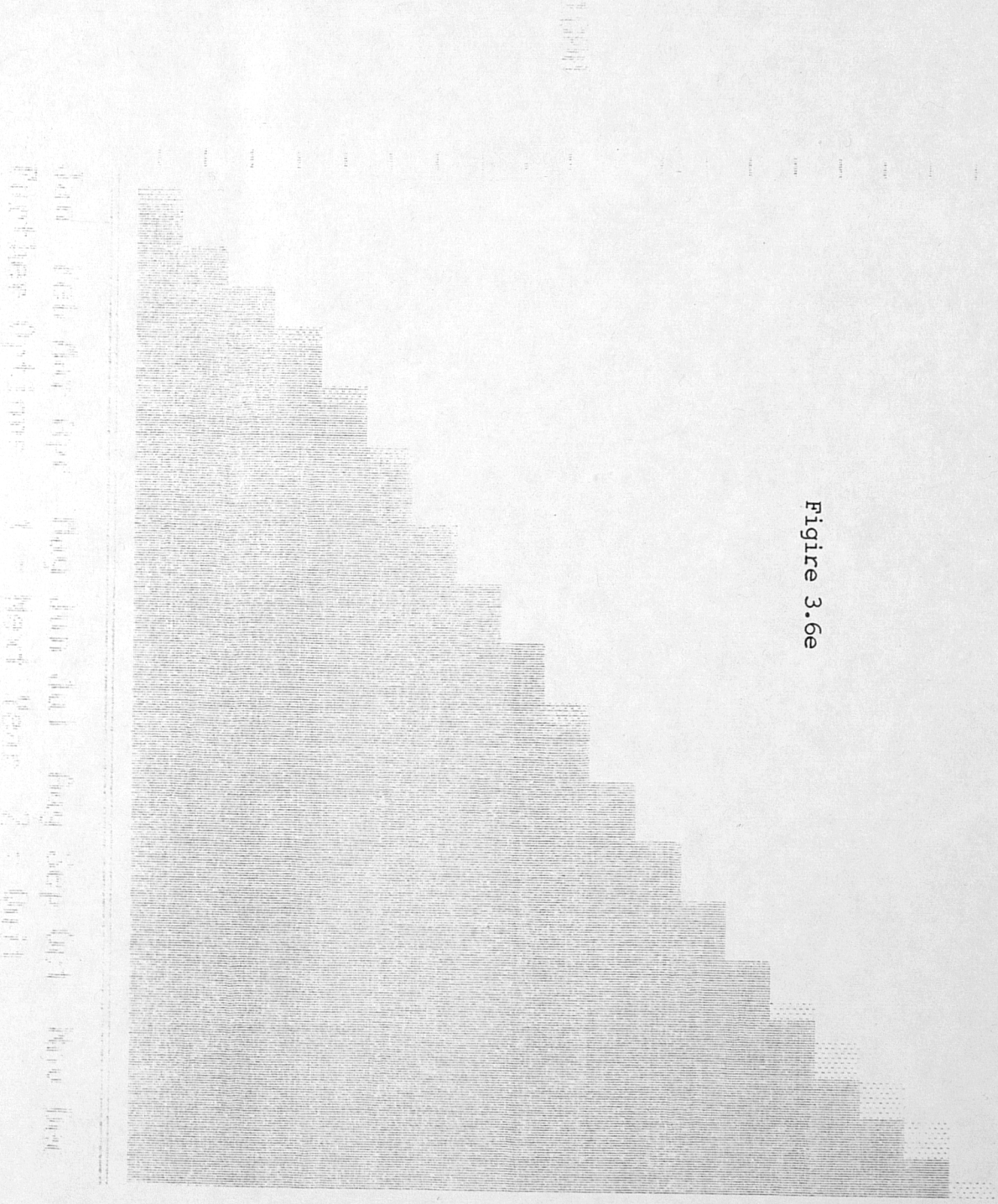


Figure 3.6d



Figire 3.6e

November in the first year. Comparing the second and third years throws up nothing as dramatic as this. Until later on in the year, both cumulative graphics barely differ, until in the closing months the second year slightly exceeds the third.

3.3.3 Results

An important observation to be made here concerns the discrepancy of the first year with the other two. If, for example, predictions had been made for the second year based solely on the first, extremely poor results would have been obtained, with crime rates for April and November being greatly over-anticipated. Officers would be expecting a massive increase in household burglaries which simply did not occur. Even in the third year, if forecasts were made based on averaging the first two years, the effect of the "spikes" in year one would still be noticeable.

The problem here is that, although certain seasonal patterns do occur in the data, some phenomena are not seasonal. In this case, it would seem that the rates in certain parts of the first year of study were something of a "one-off". This highlights the importance of the ideas put forward in the introductory chapter. In order to predict crimes successfully, another input is required to work alongside the analysis of past data patterns.

The implication here, in terms of prediction method is that the incorporation of seasonal variation into a prediction model may be of

dubious benefit. Since there are currently only three years of data available, accurate seasonal average levels are currently hard to measure. As discussed earlier, a spurious high level on a given week may have to much leverage. If data were available over a longer time period, then this may be less of a problem. In this case, it may also be possible to decompose the pattern in the frequency domain.

In addition to this, there are certain seasonal effects that, although roughly occurring at the same time each year, do not exactly match week-for-week. For example, Easter varies within a six week period. It may be better to incorporate such phenomena into the prediction system via the human interface, rather than by time series analysis of past data.

The findings of this research suggest that, due to the large datasets required for calibration, and also the unreliability of certain seasonal models, seasonal analysis is not of great benefit to a crime prediction system. This is not to say, however, that there is no benefit at all in performing analysis of this sort. As a historical pattern analysing technique in its own right, use may be made. For example, it has been identified here that in 1984, in April to May, a surprisingly large number of household burglaries occurred. This may draw attention to some phenomena that happened around this time, or lead police officers to examine crime records from around this time.

From consideration of this, some phenomena causing a massive increase in household burglaries may be identified by police officers. Having

been made aware of this, their subjective input into the system may be used next time similar phenomena occurs. Thus, although examination of time series may not directly contribute to the predictor techniques, it may be of use in a "second order" sense.

In addition to identifying the "outlying periods" as times when crime patterns may be of interest, in a subjective sense, the effects that the summer months have on burglary rates have been informally identified. A possible explanation for the general tendency for counts to be lower over the mid-summer period may be found in terms of theories of defensible space (Newman, 1972) or opportunist models (Mayhew, 1974). There are clearly greater hours of daylight during these months, and a a consequence of this, areas of some neighbourhoods that may be badly illuminated in winter are more visible to the inhabitants. This may provide less opportunities for potential burglars during the lighter months.

Finally, the "crime count target" idea must not be ignored, as a tool for monitoring the success of crime prevention (in this case household burglary) over the current year. In summary, although seasonal analysis of the crime counts has not proved to be a very powerful tool when used in the context of automated crime prediction there are other aspects of the analysis of crime patterns, to which this explanatory analysis has drawn attention, which are aided by this type of technique.

3.4 Cluster Analysis of Modus Operandi Data

Until this point, consideration of household burglary data has been mainly in the context of space or time. These are important factors, both in the identification of areas at risk, and of forecasting, but it may be of use to classify household burglaries in terms of other characteristics, in particular the methods used, and items taken. This may lead to some evidence relating to modes of behaviour by offenders; it may even identify the characteristic work of individual offenders, or of gangs, if the analysis is carried out at a sufficiently high level of geographical resolution. Classification may be performed in this way using methods of Cluster Analysis (Everitt, 1984). Multivariate data from several individual cases is examined, and patterns are sought. The end result is a classification of each individual item into a group, or cluster, all of whose elements share common characteristics. In this section, then, a pilot cluster analysis is to be performed on the crime data described in chapter 2. Since the analysis is intended to classify according to the activities of the offenders only, variables relating to space and time will be ignored. The list of variables is given in table 3.7.

3.4.1 Method

Any review of cluster analysis techniques (for example Everitt, 1984) will list a wide variety of options. Before any analysis is actually performed on this data, a suitable algorithm must be selected. It is important to note here that all of the variables to describe each burglary are qualitative, and hence non-numeric. Thus, any splitting technique should not be of the form ($x>a$ vs $x<a$). It may be useful also if

Table 3.7

Format Of The Household Burglary Data Subset Used in Cluster Analysis

Variable Name	Column(s)	Description Of Variable
ME	16	Method of Entry F=Forced B=Break In I=Insecure Building D=Drilling O=Other
PE	17	Point of Entry D=Door W=Window O=Other
DE	18	Direction of Entry F=Front S=Side B=Back R=Roof O=Other
STOLEN	19-24	Items Stolen Up to six out of C = Colour Tv H = Hi-Fi B = B/W Tv J = Jewellery V = Video Recorder E = Electrical Goods K = Cash F = Food X = Cheques D = Drink L = Clothes P = Personal Goods G = Furniture O = Ornaments M = Camera T = Tools Q = Other Or A = Attempt (No items Stolen)

division is performed hinging on single variables (monothetic division). In this way, if an algorithm is used that performs the "most important" split first, and then the next most powerful, and so on, then noting on which variable each split is made leads to substantive interpretation, in the sense that the major variational components in the characteristics of household burglary may be stated as occurrence or non-occurrence of particular behavioural traits of offenders.

Thus, the clustering algorithm should be based on discrete, non-ordered variables, and on a monothetic division technique. At this point, another problem has to be taken into account. Clustering algorithms of this sort often do not specify a "stopping rule". That is, no formal mechanism exists determining when categorisations are sufficiently homogenous to justify no further subdivision. If such a rule is not applied, there is danger of imposing spurious structure on the data. However their is usually some measure of "goodness of split" index available, and an arbitrary limit could be attached to this since the splitting is performed on a "strongest division first" basis, when the splitting only causes small changes in the index, there is little point in further divisions. Alternatively, the size of group could be monitored, until categories become reasonably small.

One suitable algorithm would be based on the information gain statistic:

$$2\Delta I = I_{p+q} - I_p - I_q$$

where $I_p = M n \log(n_p) - \sum_j f_{jp} \log(f_{jp}) + (n_p - f_{jp}) \log(n_p - f_{jp})$

n_p = size of cluster p

$f_{jp} = \sum_{i \in p} x_{ij}$ $\quad x_{ij} = 0/1$ binary data

This measures the overall increase in homogeneity when considered in terms of two separate sets split by a given variable, as opposed to one set containing both variables. Firstly, the entire data set is split two ways, according to scoring highest on the above scale defined above. After this, one of the subsets is split. The subset to be split offers the greatest gain in information. This process continues until either the information gain falls below a given limit, or the size of the categories becomes small. Note also that, after the first division, divisions do not necessarily apply across the entire dataset. Thus, after the dataset has had its initial division performed performed, each of its subsets will be further subdivided independently.

Although this type of analysis is useful, and simple to interpret, since divisions are always made on the basis of a single variable, it is subject to certain problems. Since the categorisation structure is essentially tree-based, all later categorisations will depend on the outcome of earlier ones. Thus, if a spurious effect occurs early on in the analysis, the errors due to this are propagated through all subsequent categorisations. To overcome this, other categorisation methods exist, based on the concept of dynamic reallocations (Everitt, 1984). In this instance, observations at any stage of analysis may be moved from one category into any other, in order to optimise some "goodness of categorisation" parameter. In this case, early mistakes may be remedied later on, since any categorisation made at some point has some chance of being re-categorised. It is generally thought that this type of analysis is less susceptible to distortion due to outliers, or unusual observations, within the dataset.

However, this does not come without a price. The categorisations no longer have the simple definitions obtained with monothetic division, and are considerably harder to interpret substantively. They are also considerably harder to assign new cases to when attempting a classification of subsequent data based on the results of the initial studies.

It is therefore proposed to perform a two-tier analysis here. Firstly, the more easily interpretable monothetic division technique (based on entropy maximisation) will be performed. The results of this will then be compared to those of a dynamic reallocation algorithm. This acts as a form of "verification" clearly similar results in both cases suggest that the interpretation of the first analysis in substantive terms is reasonable (assuming the reliability of the second level). However, widely differing results suggest that the first, monothetic method, may have greatly distorted categorisation early on due to a spurious observation.

3.4.2 Implementation

Methods of cluster analysis have been proposed in the previous section, but, as yet little consideration has been given to practical aspects. Firstly, consider the computing aspects of the problem. There exists a package, CLUSTAN, which is specifically designed to perform cluster analysis as required in this study. It is capable of analysing either categorical or ordered data. In the case of categorical variables, these must be coded as binary. Thus, in the case of categorical variables with more than two classes, dummy variable encoding must be performed.

Thus, the variables in table 3.7 are re-expressed in the variable set given in table 3.8. In particular, the original list of items taken must be recorded as a set of binary variables of the form " Colour TV taken" etc. These variables begin at number 14, dividing the variable set into two main categories, "means of entry" and "items taken". Note that as variables (1-3), (4-7) and (8-12) are dummy variable interpretations of the variables 1, 2 and 3 in table 3.7. Since only one of each of the sets (1-3), (4-7) and (8-12) can be assigned a logical "true" value, and the remaining elements must be "false" there is a certain amount of correlation arising by design. However, if this is identified at the outset, effects due to it may be allowed for when interpreting results.

Finally, a dataset size constraint is applied by the CLUSTAN package. Owing to the memory addressing and time constraints at the time the package was written, CLUSTAN is restricted to datasets not exceeding 999 cases. The full year of detailed data available, containing information about modus operandi, consists of about 1800 observations. Thus, a random selection of 999 cases must be made from the original database before cluster analysis is performed. In summary, then, the original dataset must be narrowed down to 999 items by random selection and these items must then be recoded in dummy binary variables format. At this stage, the CLUSTAN package may be used to perform both monothetic division classification, and dynamic reallocation based methods.

Table 3.8

Dataset For Cluster Analysis Expressed As Binary Variables

Variable Number	Variable Name
1	Entry By Force ?
2	Entry by breaking in ?
3	Entry due to insecurity ?
4	Entry by drilling ?
5	Entry by Other Means ?
6	Entry through door ?
7	Entry through Window ?
8	Entry by other means ?
9	Entry at front ?
10	Entry at side ?
11	Entry from rear ?
12	Entry from roof ?
13	Entry from another point ?
14	Colour Tv Stolen ?
15	HiFi Stolen ?
16	B/W Tv Stolen ?
17	Jewellery Stolen ?
18	Attempted burglary ?
19	Video Recorder Stolen ?
20	Electrical Goods Stolen ?
21	Cash Stolen ?
22	Food Stolen ?
23	Cheques Taken ?
24	Drink Taken ?
25	Clothes Taken ?
26	Personal goods taken ?
27	Furniture Taken ?
28	Ornaments Taken ?
29	Camera Taken ?
30	Tools Taken ?
31	Other Items Taken ?

3.4.3 Results

The results of the monothetic division algorithm are given in table 3.9. After four levels of subdivision, the group sizes were between 20 and 40 with a few larger ones, which seems reasonable. Due to some observations having missing values for some of the variables, the final number of items processed is slightly less than 999. The first division was made on variable 6, which indicates whether entry was through a door or by some other means. Note that, in the case of the "other means" subdivisions the first division here is by variable 7, "entry through window".In all of the divisions carried out to the third level, the variables have come from the "means of entry" category as opposed to the "items taken".

The results of the reallocation based clustering technique are listed in table 3.10. Again, missing values slightly reduce the final total of cases processed. In this case, the conditions for membership of each class are considerably more complex, and harder to interpret than those of the monothetic technique. There is no obvious structure in the classification. Although, eventually, there are roughly the same number of categories in each case, and the range of sizes of observations in the category are roughly equivalent, the classification from the first algorithm is done on the basis of four binary divisions, whilst in the second case, as many as 25 binary variables need to be considered.

However, it may still be noted that those binary variables concerned with method of burglary rather than items stolen appear most frequently.

Table 3.9
Results Of Monothetic Division Cluster Analysis

Group No.	Splitting Variables '+' = true '-' = false				Size Of Group
1	+6	+1	+9	+14	35
2	+6	+1	+9	-14	136
3	+6	+1	-9	+21	22
4	+6	+1	-9	-21	55
5	+6	-1	+9	+2	24
6	+6	-1	+9	-2	86
7	+6	-1	-9	+21	29
8	+6	-1	-9	-21	67
9	-6	+7	+9	+19	46
10	-6	+7	+9	-19	143
11	-6	+7	-9	+2	44
12	-6	+7	-9	-2	167
13	-6	-7	+5	(+/-13)	37
14	-6	-7	-5	+4	23
15	-6	-7	-5	-4	41

Since group 13 and the original 14 were small, they have been merged with each other. In terms of splitting, the groups are nearest neighbours.

Table 3.10
Results Of Reallocation Algorithm Cluster Analysis

Group no.	Splitting Variables '+' = true / '-' = false							Size Of Group
1	+18	-19	-23	-21	-4	-25	-10	38
	-7	-17	-11	-20	-12	-22	-27	
	-24	-28	-26	-30	-3	-29	-14	
	-31	-15	-16	-24				
2	+5	+7	-1	-12	-30	-8	-11	46
	-10	-22	-18	-6	-2	-4	-3	
3	+14	-7	-3	-18	-4	-26	-10	77
	-2	-8	-11	-12	-24	-23		
4	+7	+14	-1-	-4	-3	-24	-12	72
	-8	-26	-11	-22	-30	-6	-18	
	-16							
5	+5	-10	-24	-23	-11	-25	-1	36
	-30	-7	-29	-4	-2	-3	-22	
6	+1	+6	+9	-12	-18	-7	-5	119
	-13	-2	-11	-8	-14	-4	-10	
	-3							
7	+2	+6	-24	-26	-8	-7	-10	50
	-3	-12	-18	-4	-1	-11	-5	
	-22							
8	+7	-24	-3	-5	-10	-18	-12	116
	-9	-4	-8	-30	-6	-14	-22	
9	-10	-24	-11	-18	-27	-12	-29	65
	-22	-7	-1	-6				
10	+7	+9	-18	-13	-11	-12	-10	122
	-30	-8	-14	-4	-6	-3	-5	
	-22							
11	+18	+7	-2	-22	-26	-8	-20	44
	-4	-24	-10	-17	-11	-19	-12	
	-21	-27	-23	-28	-25	-30	-6	
	-3	-29	-14	-31	-15	-16		
12	+6	-4	-11	-9	-7	-18	-10	76
	-30	-22	-2	-8	-24			

The main variable seen to be having greater influence here than in the first analysis is that of "no items stolen" in the category of "outcome" variables. This appears in nine out of the twelve categories generated by this analysis.

3.4.4. Discussion

Firstly, some attempt must be made to explain the categories obtained by the initial monothetic analysis. For each class in table 3.9, a verbal description is given in table 3.11. As noted earlier, most of the distinctions are based on methods of entry rather than items stolen. This is perhaps a reasonable outcome: although potential offenders have control over their modus operandi, except in a few cases they are unlikely to be aware of the exact contents of a dwelling. Thus, definite patterns in means of entry, perhaps due to local trends or even to traits of particular individuals may become evident in the analysis. However, since the items taken are more likely to vary randomly according to the internal layout, and obviously the contents of households, it is less likely that there will be strong patterns in this type of data. A few items likely to be found in several houses, possibly in easily predictable positions, may be deliberately sought out. This may explain the appearance of a few items in the classification (ie. video recorders, cash).

The results of the reallocation algorithm analysis will now be considered in a similar manner. The categories from this analysis are described verbally in table 3.12. The descriptions here are more complex, and do

Table 3.11

Descriptions of Monothetic Division Groupings

1	Force front door, take Tv.
2	Force front door, no Tv taken.
3	Force door other than front, take cash.
4	Force door other than front, dont take cash.
5	Break glass on front door.
6	Insecure front door.
7	Other means than force through non-frontal door. Take cash.
8	Other means than force through non-frontal door. Dont take cash.
9	Front window entry taking video.
10	Front window entry not taking video.
11	Non-frontal window break-in
12	Non-frontal window, not breaking in.
13	Entry not through door or window not by drilling breaking or force.
14	Drill entry not through window or door.
15	Break-in or force, not through window or door.

Table 3.12
Descriptions Of Reallocation Algorithm Groupings

1	Attempt, Not through window
2	Front window entry, not by force, drilling or breaking
3	Colour Tv taken, but no personal goods. Forced door entry.
4	Colour Tv taken via forced or broken front window.
5	Front or roof entry, not by drilling , force or breaking.
6	Force front door. No Colour Tv taken.
7	Break front door. No food or drink taken.
8	Force or break front or rear window.
9	Front or rear entry, not by door or window. No force. No furniture, drink or tools stolen.
10	Force or break front window. No food or tools stolen.
11	Attempt via window
12	Forced or inseecure door entry from roof. No food or drinks taken.

not fully reflect the binary attribute classifications defining each class. However, there is a reasonable correspondence between the outcome of each of the two analyses. There is a similar range in size and similar number of classes. Also, method of entry variables are prominent, in the definition of classes. Thus, the simpler classifications arrived at in the first analysis seem reasonable. Given this, there is another factor evident in this analysis that does not occur in the former. The outcome of having no items stolen - ie., the burglary being classified as an attempt only, appears in the categorisation frequently.

This may be an important factor. This split could represent the level of security to which homes are protected. An unsuccessful burglary may be the product of fitting window locks (in the case of category 11) or secure doors (Category 1). Also, this sheds some light onto other categories, such as number 12, in which an insecure point of entry was used as the means of entrance.

In conclusion, this analysis has identified some important characteristics of household burglaries, which may be used to provide a classification. Clearly, the simpler option offered by the monothetic algorithm seems preferable to the more complex reallocated result; it is easier to interpret, and would be simpler to implement on a computer-based classification program to be applied to future data.

However, the second analysis identified the attempt/successful burglary criterion as also being of importance. It is therefore recommended that any classification of burglaries have categories corresponding to those

from the monothetic division algorithm, but with the addition of two further categories, "attempt through window" and "attempt through door" added to the analysis.

3.5 Conclusion of Chapter

At the start of this chapter it was pointed out that it was not necessarily intended to incorporate the methods used here directly into any final crime pattern analysis system and that the purpose of these investigators were mainly exploratory. Certain conclusions have been drawn from each of the three studies. In particular, the time-of-day study not only yielded interesting results, particularly with the difference in daily patterns, but also gave a new quantitative technique, which was required as a result of the nature of the data presented for analysis. It is also hoped that some insight into criminal behaviour may have been gained from these analyses. Certain patterns discovered here may enable some empirically-based inferences to be made about the experience of criminals. Another "by product" is that certain analyses performed here were in the form of microcomputer programs of the type that were suggested for the crime pattern analysis software proposed in chapter one. Thus, although the techniques here would not become part of the "mainstream" system, they could be included as subsidiary facets of the final system.

Finally, then, it is hoped that this exploratory chapter in crime pattern analysis techniques, although not yielding results that may directly be incorporated into a prediction system or mapping system, may have

provided some insight into the processes behind the data, and may be of subsidiary value when building the main model of the study. In addition to this, they also strengthen the original belief that any operational crime forecasting system should be based on the most basic space and time data.

LISTINGS FOR CHAPTER 3

```
      PROGRAM TMRISK
C
C
C     ******** Listing 3.1 ******
C
C                                                    C
C    Kernel Estimator For Police Interval Data       C
C                                                    C
C                                                    C
      REAL*4 CUT
      INTEGER*4 N, TCKSUM, ITEMS, ITCUT
      INTEGER*4 DVEC1(2100), DVEC2(2100), DVEC3(2100), DVEC4(2100)
      INTEGER*4 WEEK(2100), DAY(2100)
      CHARACTER CSTR*60, INFILE*12
      LOGICAL PLOT, NOCUT, SEASDV, WEEKDV
      COMMON /STOR/ DVEC1, DVEC2, DVEC3, DVEC4, WEEK, DAY, ITEMS,
     1              ITCUT, NOCUT, PLOT
      N = 0
      TCKSUM = 0
C
C  Read the command string
C
      CALL GETCOM(CSTR)
C
C  Check for seasonal and week / weekend divides
C
      SEASDV = (INDEX(CSTR,'SEASONS')  .NE. 0)
      WEEKDV = (INDEX(CSTR,'WEEKENDS') .NE. 0)
      PLOT   = (INDEX(CSTR,'PLOT')     .NE. 0)
C
C  Check for cutoff points
C
      NOCUT = ( INDEX(CSTR,'CUT') .EQ. 0 )
      IF (.NOT. NOCUT) THEN
        WRITE (6,'(21H&Enter cutoff time > )')
        READ (5,*) CUT
        CUT = CUT * 2
        ITCUT = INT(CUT)
        WRITE (6,*)
      END IF
C
C  Assign the data file to unit 4.
C
      IND1 = INDEX(CSTR,'$')
      IF (IND1 .EQ. 0) THEN
        WRITE (6,'(A)') ' Improper control string -- no data file'
        STOP 1
      END IF
      IND2 = INDEX(CSTR(IND1:),' ')
      INFILE = CSTR(IND1+1:IND2)
      OPEN(4, FILE=INFILE)
C
C  Read everything in
C
      WRITE (6,*) ' Data entry in progress ... '
      WRITE (6,*) ' Reading formatted file ... '
      WRITE (6,*)
      ITEMS = 1
   91 READ (4,'(I2,I1,T4,2I2,T13,I3,I2)',END=90)
     1WEEK(ITEMS), DAY(ITEMS),
     2DVEC1(ITEMS), DVEC2(ITEMS), DVEC3(ITEMS), DVEC4(ITEMS)
```

```
C
C  Convert day of week to weekday/weekend indicator
C
      IF (DAY(ITEMS) .LE. 5) THEN
        DAY(ITEMS) = 1
      ELSE
        DAY(ITEMS) = 2
      END IF
C
C  Convert week counter to seasonal indicator 1=winter --> 4=autumn
C
      IF (WEEK(ITEMS) .GT. 8) THEN
        WEEK(ITEMS) = (WEEK(ITEMS)-8)/13 + 2
        IF (WEEK(ITEMS) .EQ. 5) WEEK(ITEMS) = 1
      ELSE
        WEEK(ITEMS) = 1
      END IF
      IF (MOD(ITEMS,100) .EQ. 0)
     1  WRITE (6,'(''+At record '',I4)') ITEMS
      ITEMS = ITEMS + 1
      GO TO 91
   90 ITEMS = ITEMS - 1
      WRITE (6,*)
C
C  Split up as appropriate
C
      IF (SEASDV) THEN
        IF (WEEKDV) THEN
           CALL KERNL(1, 1, 'Weekdays Winter')
           CALL KERNL(2, 1, 'Weekends Winter')
           CALL KERNL(1, 2, 'Weekdays Spring')
           CALL KERNL(2, 2, 'Weekends Spring')
           CALL KERNL(1, 3, 'Weekdays Summer')
           CALL KERNL(2, 3, 'Weekends Summer')
           CALL KERNL(1, 4, 'Weekdays Autumn')
           CALL KERNL(2, 4, 'Weekends Autumn')
         ELSE
           CALL KERNL(0, 1, ' Winter')
           CALL KERNL(0, 2, ' Spring')
           CALL KERNL(0, 3, ' Summer')
           CALL KERNL(0, 4, ' Autumn')
         END IF
       ELSE
         IF (WEEKDV) THEN
           CALL KERNL(1,0,' Weekdays')
           CALL KERNL(2,0,' Weekends')
         ELSE
           CALL KERNL(0,0,' All')
         END IF
       END IF
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE KERNL(SPLIT1, SPLIT2, TITLE)
C
C  Kernel estimation subroutine
C
      CHARACTER*(*) TITLE
      REAL*4 KERNEL(0:47), ICRMNT, IGRAL, MAXKRN
```

```
      INTEGER*4 FTICK, ITICK, HRFND, MNFND, HRINT, MNINT, HOUR, MIN
      INTEGER*4 TIME, TCKSUM, LEN, ITEMS, ITCUT
      INTEGER*4 DVEC1(2100), DVEC2(2100), DVEC3(2100), DVEC4(2100)
      INTEGER*4 USED, WEEK(2100), DAY(2100), SPLIT1, SPLIT2
      CHARACTER CSTR*60, INFORM*12, INFILE*12, DUMMY*40
      LOGICAL PLOT, NOCUT
      COMMON /STOR/ DVEC1, DVEC2, DVEC3, DVEC4, WEEK, DAY, ITEMS,
     1              ITCUT, NOCUT, PLOT
C
C  Empty the kernel estimator
C
      DO 100 I = 0, 47
  100   KERNEL(I) = 0.0
C
C   The Kernel is empty -- Start to build it
C
      USED = 0
      TCKSUM = 0
      WRITE (6,*) ' Building Kernel estimate ...'
      WRITE (6,*)
      DO 110 ITEM = 1, ITEMS
  120   HRFND = DVEC1(ITEM)
        MNFND = DVEC2(ITEM)
        HRINT = DVEC3(ITEM)
        MNINT = DVEC4(ITEM)
C
C  Convert into 48-unit day : called 'ticks'.
C
        FTICK = HRFND*2 + MNFND/30
        ITICK = HRINT*2 + MNINT/30
        IF (SPLIT1.EQ.0 .OR. SPLIT1.EQ.DAY(ITEM)) THEN
          IF (SPLIT2.EQ.0 .OR. SPLIT2.EQ.WEEK(ITEM)) THEN
            IF (ITICK.LT.ITCUT .OR. NOCUT) THEN
C
C   We now have how long before found,  and when found
C   Make this into start of interval + length
C
              FTICK = FTICK - ITICK
  40          IF (FTICK .LT. 0) THEN
                FTICK = FTICK + 48
                GO TO 40
              END IF
C
C  Add the kernel to the overall estimate
C
              TCKSUM = TCKSUM + ITICK
              ICRMNT = 1.0/FLOAT(1+ITICK)
              DO 130 TIME = 0, ITICK
                IDX = MOD(FTICK + TIME, 48)
  130           KERNEL(IDX) = KERNEL(IDX) + ICRMNT
              USED = USED + 1
            END IF
          END IF
        END IF
C
C  Report the status every 100 items --- stops 'long silences' on VDU
C
        IF (MOD(ITEM,100).EQ.0)
     1    WRITE(6,'(11H+At record  ,I4,12H Cases used ,F4.1,1H%)')
     2    ITEM, 100.0 * FLOAT(USED)/ITEM
  110 CONTINUE
```

```
C
C  Re-scale so that 48-element kernel array sums to unity.
C
      IGRAL = 0.0
      DO 200 I = 0, 47
  200   IGRAL = IGRAL + KERNEL(I)
      MAXKRN = 0.0
      DO 210 I = 0, 47
        KERNEL(I) = 100.0 * KERNEL(I)/IGRAL
        IF (KERNEL(I) .GT. MAXKRN) MAXKRN = KERNEL(I)
  210   CONTINUE
C
C  48 - point kernel Estimator lies in array KERNEL
C
      HOUR = 0
      MIN  = 0
C
C  Graph or tabulate kernel estimator
C
      IF (PLOT) THEN
C
C  Graph option
C
        CALL GRAPH(KERNEL,MAXKRN)
        WRITE (DUMMY,'(''Number  Of Cases           = '',I4)') USED
        CALL PUTTXT(21,1,DUMMY)
        WRITE (DUMMY,'(''Average Interval (Hours)   = '',F7.3)')
     1   FLOAT(TCKSUM)/(USED * 2)
        CALL PUTTXT(21,2,DUMMY)
        CALL PUTTXT((80-LEN(TITLE))/2, 0, TITLE)
      ELSE
C
C  Table option
C
        WRITE (6,'(A)') TITLE
        WRITE (6,'(1H )')
        DO 140 I = 0, 47
        WRITE (6,'(1H&,I2.2,1H:,I2.2,4X,F6.2,1H%,4H    )')
     1   HOUR, MIN, KERNEL(I)
        IF (MOD(I+2 1,3) .EQ. 0) WRITE(6,'(1H )')
        MIN  = MIN + 30
        IF (MIN .EQ. 60) THEN
          MIN = 0
          HOUR = HOUR + 1
        END IF
  140   CONTINUE
      WRITE (6,'(15H Average gap = ,F7.3,5H Hrs.)')
     1  FLOAT(TCKSUM)/(USED * 2)
      WRITE (6,'(15H Items used  = ,I3)') USED
      END IF
      CALL KEY
      CALL MODE(3)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE GRAPH(KERNEL,MAXKRN)
C
C  Draw bar graph of kernel estimate on screen
C
```

```
      REAL*4 KERNEL(0:47), MAXKRN
      INTEGER HEIGHT
      CHARACTER*4 LABEL
      CALL MODE(16)
C
C  Underline Bar Graph
C
      DO 100 I = 16, 63
  100   CALL CPUT(I, 21, 223, 15)
      DO 105 J = 3, 21
  105   CALL CPUT(15, J, 196, 13)
C
C  Enter the axes
C
      CALL PUTTXT(34,24,'Hour of Day')
      CALL PUTTXT(1,7,'Percentage')
      CALL PUTTXT(1,8,'  of all  ')
      CALL PUTTXT(1,9,'Household ')
      CALL PUTTXT(1,10,'Burglaries')
      DO 110 I = 16, 34, 2
        CALL CPUT(I, 22, 48, 7)
  110   CALL CPUT(I, 23, 48+(I-16)/2 , 7)
      DO 120 I = 36, 54, 2
        CALL CPUT(I, 22, 49, 7)
  120   CALL CPUT(I, 23, 48+(I-36)/2, 7)
      DO 130 I = 56, 62, 2
        CALL CPUT(I, 22, 50, 7)
  130   CALL CPUT(I, 23, 48+(I-56)/2, 7)
C
C  Now for the hard stuff
C
C
C  First,  sort out a "clean" scale
C
      IF (MAXKRN .GT. 10) THEN
        MAXKRN = INT(MAXKRN)/10
        MAXKRN = 10.0 * (MAXKRN + 1)
      ELSE
        MAXKRN = 1.0 + INT(MAXKRN)
      END IF
      WRITE (LABEL,'(F4.1)') MAXKRN
      CALL PUTTXT(11,3,LABEL)
      WRITE (LABEL,'(F4.1)') MAXKRN/2
      CALL PUTTXT(11,12,LABEL)
      CALL PUTTXT(11,21,' 0.0')
C
C  Now plot it
C
      DO 140 I = 0, 47
        IMAP = INT((KERNEL(I) / MAXKRN) * 36.0 + 0.49)
        HEIGHT = IMAP / 2
C       ICOL = $A1
C       IF (MOD(I,2) .EQ. 0) ICOL = $92
        ICOL = 1
        IF (I.GT.12.AND.I.LT.40) ICOL = 14
        IF (MOD(IMAP, 2) .EQ. 1)
     1    CALL CPUT(16+I, 20-HEIGHT, 220, ICOL)
        IF (HEIGHT .GE. 1) THEN
          DO 150 J = 1, HEIGHT
  150     CALL CPUT(16+I, 21-J, 219, ICOL)
        END IF
```

```
  140   CONTINUE
        RETURN
        END
C
C*************************************************************************
C
        SUBROUTINE CPUT(XTL, YTL, CHAR, ATTR)
C
C   Put a character on the screen with given attribute
C
        INTEGER*4 XTL, YTL, CHAR, ATTR
        INCLUDE 'A:SYSREG.FOR'
        AH = 2
        BH = 0
        DL = XTL
        DH = YTL
        CALL SYS2(16,SYSREG)
        AH = 9
        AL = CHAR
        BL = ATTR
        CX = 1
        CALL SYS2(16,SYSREG)
        RETURN
        END
C
C*************************************************************************
C
        SUBROUTINE KEY
C
C  Wait for a key to be pressed
C
        INCLUDE 'A:SYSREG.FOR'
        AH = 7
        CALL SYS1(SYSREG)
        RETURN
        END
```

```
      PROGRAM SEASON
C
C         ********* Listing 3.2 **********
C
C
C  Seasonal analysis of crime rates --- draws crime rate graphs
C  and cumulative graphs overlaid year by year
C
      INTEGER COUNT(52), CHOICE, MAXMUM, HEIGHT, LCOUNT(52)
      INTEGER HGHT1, HGHT2
      CHARACTER INFILE*30, DUMMY*6
C
C  Attach the data file to unit 4
C
      CALL GETCOM(INFILE)
      OPEN(4, FILE=INFILE)
C
C  Empty the screen -- menu up for Cumulative/Simple display
C
      CALL MODE(3)
      CALL PUTTXT(25, 2,'Seasonal Crime Rate Analysis')
      CALL PUTTXT(25, 6,'Select display mode :-')
      CALL PUTTXT(26,10,'1 == Simple Seasonal Curve')
      CALL PUTTXT(26,12,'2 == Cumulative Comparison')
  100 CALL KEYGET(CHOICE)
      IF (CHOICE .NE. 49 .AND. CHOICE .NE. 50) GO TO 100
C
C  Choice is now made -- now put the approriate graph on VDU
C
      IF (CHOICE .EQ. 49) THEN
C
C  Simple seasonal curve
C
  160   CALL MODE(3)
C
C  Get the count rates
C
        MAXMUM = 0
        DO 105 I = 1, 52
          READ (4,*,END=170) COUNT(I)
          IF (COUNT(I) .GT. MAXMUM) MAXMUM = COUNT(I)
  105   CONTINUE
C
C  Now get a 'good' scale
C
        CALL RESCAL(MAXMUM)
C
C  Plot it
C
        DO 110 I = 1, 52
          HEIGHT = 20*(COUNT(I)/FLOAT(MAXMUM))
          DO 120 J = 22 - HEIGHT, 22
            CALL CPUT(I+14, J, 178, 13)
  120     CONTINUE
  110   CONTINUE
C
C  Plot the axis
C
        DO 130 I = 15, 66
  130     CALL CPUT(I, 22, 205, 14)
        DO 140 I = 2, 22
```

```
 140     CALL CPUT(14,I, 196, 7)
C
C  Plot the labelling
C
        CALL PUTTXT(15,23,
     1    'Jan  Feb Mar Apr  May Jun Jul  Aug Sep Oct  Nov Dec')
        WRITE (DUMMY,'(I6)') MAXMUM
        CALL PUTTXT(4,2,DUMMY)
        WRITE (DUMMY,'(I6)') MAXMUM / 2
        CALL PUTTXT(4,12,DUMMY)
        CALL PUTTXT(11,1,'Weekly Counts')
        CALL PUTTXT(21,0,'Seasonal Household Burglary Variation')
C
C  Put up further menu options on bottom line
C
        CALL PUTTXT(15,24,'Further Options   1 = Next Year  2 = Quit')
C
C  Get new menu choice
C
 150    CALL KEYGET(CHOICE)
        IF (CHOICE .NE. 49 .AND. CHOICE .NE. 50) GO TO 150
        IF (CHOICE .EQ. 49) GO TO 160
      ELSE
C
C  Cumulative data analysis -- overlays two years worth of data
C
C
C  First get the data
C
        DO 200 I = 1, 52
          READ (4,*,END=170) LCOUNT(I)
 200    CONTINUE
 370    DO 210 I = 1, 52
          READ (4,*,END=170)  COUNT(I)
 210    CONTINUE
C
C  Now make it cumulative
C
        DO 220 I = 2, 52
          LCOUNT(I) = LCOUNT(I-1) + LCOUNT(I)
 220      COUNT(I)  = COUNT(I-1)  +  COUNT(I)
        IF (LCOUNT(52) .GT. COUNT(52)) THEN
          MAXMUM = LCOUNT(52)
        ELSE
          MAXMUM =  COUNT(52)
        END IF
C
C  Plot it
C
        CALL RESCAL(MAXMUM)
        CALL MODE(3)
        DO 230 I = 1, 52
          HGHT1 = 20 * (LCOUNT(I) / FLOAT(MAXMUM))
          HGHT2 = 20 * ( COUNT(I) / FLOAT(MAXMUM))
          IF (HGHT1 .GE. HGHT2) THEN
            DO 240 J = 22-HGHT1, 22-HGHT2
 240          CALL CPUT(14+I, J, 178, 7)
            DO 250 J = 22-HGHT2, 22
 250          CALL CPUT(14+I, J, 219, 7)
          ELSE
            DO 260 J = 22-HGHT2, 22-HGHT1
```

```
 260         CALL CPUT(14+I, J, 176, 7)
           DO 270 J = 22-HGHT1, 22
 270         CALL CPUT(14+I, J, 219, 7)
         END IF
 230   CONTINUE
C
C  Plot labels
C
       CALL PUTTXT(15,23,
    1    'Jan  Feb Mar Apr  May Jun Jul  Aug Sep Oct  Nov Dec')
       WRITE (DUMMY,'(I6)') MAXMUM
       CALL PUTTXT(4,2,DUMMY)
       WRITE (DUMMY,'(I6)') MAXMUM / 2
       CALL PUTTXT(4,12,DUMMY)
       CALL PUTTXT(22,0,'Cumulative Household Burglary Counts')
       CALL PUTTXT(25,1,'  = This Year      = Last Year')
       CALL CPUT(25,1,176,7)
       CALL CPUT(41,1,178,7)
C
C  Plot axes
C
       DO 340 I = 15, 66
 340     CALL CPUT(I, 22, 205, 14)
       DO 350 I = 2, 22
 350     CALL CPUT(14,I, 196, 7)
C
C  Put up further menu options on bottom line
C
       CALL PUTTXT(15,24,'Further Options   1 = Next Year  2 = Quit')
C
C  Get new menu choice
C
 365   CALL KEYGET(CHOICE)
       IF (CHOICE .NE. 49 .AND. CHOICE .NE. 50) GO TO 365
       IF (CHOICE .EQ. 49) THEN
C
C  Roll on another year --- reconvert to non-cumulative
C
         DO 360 I = 52, 2, -1
 360       LCOUNT(I) = COUNT(I) - COUNT(I-1)
         LCOUNT(1) = COUNT(1)
         GO TO 370
       END IF
     END IF
     GO TO 180
C
C Exeption Handling : leave program if no further data is available
C
 170 CALL MODE(3)
     CALL PUTTXT(18,13,'No further data : Press any key to continue')
     CALL KEYGET(CHOICE)
 180 STOP
     END
C
C*********************************************************************
C
     SUBROUTINE RESCAL(MAXMUM)
C
C  Subroutine to rescale graph to a 'nice' number ie 10, 20, 50 etc
C
     INTEGER SCALE, MULT, DIV, MAXMUM
```

```
      SCALE = 1
      MULT = 2
      DIV = 1
  100 IF (SCALE .LT. MAXMUM) THEN
        SCALE = SCALE * MULT
        SCALE = SCALE / DIV
        IF (MULT .EQ. 2) THEN
          MULT = 5
          DIV = 2
        ELSE
          IF (MULT .EQ. 5) THEN
            MULT = 4
            DIV = 2
          ELSE
            IF (MULT .EQ. 4) THEN
              MULT = 2
              DIV = 1
            END IF
          END IF
        END IF
        GO TO 100
      END IF
      MAXMUM = SCALE
      RETURN
      END
C
C*************************************************************************
C
        SUBROUTINE KEYGET(CODE)
C
C  Wait for a key to be pressed -- Return its code
C
        INTEGER CODE
        INCLUDE 'A:SYSREG.FOR'
        AH = 7
        CALL SYS1(SYSREG)
        CODE = AL
        RETURN
        END
```

C H A P T E R 4

ANALYSIS OF SPACE-TIME PATTERNS IN CRIME DATA

4.1 Introduction

The need to perform spatial analysis to predict spatial pattern has already been identified, both in terms of police manpower management and also in terms of local scale information for beat police officers. In order to measure, draw inference from and forecast using these patterns, some theoretical background describing the processes generating them should be considered. In addition to this, consideration must be given to crime patterns as a specific phenomena rather than modelling in terms of a general spatial process, and to the particular needs of the crime pattern analyst, so that the final model describes this particular phenomena well, and is capable of producing output in a format that will be easily understood by people working in this field. If this target is attained, it is hoped that reliable forecasting, and spatial pattern spotting techniques, will be achieved.

An important concept in defining stochastic processes over space is that of interaction. A spatial process in which the value of a realisation at any given point in space is unaffected by the values that it takes at any other point would be a trivial one. Such a process

would require no knowledge of the state of any other point in space to make predictions about a particular point. However, most real life situations are not well described by a model such as this. For example, fields with a high annual yield within a particular farming area might tend to cluster in space, due to soil properties overlapping field boundaries. Similarly, parallels may be found in many other areas of study. The factor connecting all of these processes is that, given a value of some observation at a point in space P, the expected value at points near to P will be altered (cf Tobler, 1970). This is the phenomena of spatial interaction, or spatial autocorrelation.

In describing spatial probabilistic processes modelling the occurrence of crimes, it is important to determine whether the phenomena of autocorrelation in space should be allowed for. Ideally, some form of non-parametric, exploratory testing should be carried out before explicitly modelling the occurrence of crime in space with probability distributions.

Such problems will be considered in this chapter, together with the problems of finding probabilistic models which suitably describe the observed data on household burglaries, whose collection is described in chapter 2. Further thought will be given to the ways of expressing the crimes as spatial data, together with the parallel stochastic mathematical models of the process. The crimes may be expressed both as points in space (by considering the grid references of households, or at least of their postcodes) or may be aggregated by regions, such as foot beats or grid squares. In the first

instance, data consists of a list of two dimensional points, and the hypothesis of autocorrelation under test is that existence of events at certain points in space does not affect the probability that events occur at other points nearby.

In the latter occurrence, the set of aggregated crime counts, or their densities in terms of the numbers of households within the zone of aggregation, are associated with a matrix of similarities for the set of regions. These similarities are often in terms of distance, although they need not necessarily be so. Tests are then made to see if observations for the regions correlate, and if so then to examine whether correlations correspond to the distance based similarity measures.

For predictive purposes, the second type of model may be of greater use, as it is not really possible to predict where crime will occur to the point level of resolution. Furthermore, the foot beat region is a useful administrative unit for police resource management. It is felt, however, that failure to examine point patterns could lead to certain spatial processes going undetected; Aggregated data may detect interaction on a scale as large as (and also larger than) the sizes of the foot beat region, but some processes may be wholly contained within foot beats. In the latter case, aggregation would result in the loss of all information about the process. The methodology, then, is to examine point processes first, and then, bearing in mind that the aggregate processes will be related to these, formulate the corresponding models. It is worth remembering that

although some clustering in space takes place entirely within the foot beat regions, other clusters are likely to occur on the same scale, but displaced so as to overlap zonal borderlines.

Thus in an aggregate analysis lack of apparent autocorrelation could be attributed to the fact that no clusters in a particular study sample crossed any borders. This could lead to predictive difficulty, which could be avoided if pointwise analysis were possible. Also, in the final prediction system, pointwise analysis could also be available for past data, as it may identify 'clusters' which may then be investigated for further pattern, in terms of mode-of-entry or other non-spatial data. Thus while beatwise analysis gives an overview of the system as a whole, and is useful for prediction, for individual police beat officers who may wish to investigate in greater detail smaller clusters of phenomena within there patrol area, pointwise analysis of past data may be of more use. In addition to considering crime occurrence as a process having spatial interaction and correlation, thought should also be given to the time dimension. Not only does one expect that events near in space are correlated, but also those that are close in time will interact. In order to model the process completely, a full space-time stochastic process must be considered. This may be done in several ways. Firstly an analysis of spatial interaction over several time periods may be carried out. It is possible that over different time periods, differing time-aggregated spatial patterns may become apparent. After this, more sophisticated models may be developed, with both space and time

autocorrelation, and interaction between the two aspects. These might be considered as a time series of vectors in the case of aggregated data analysis. These refinements add realism to the model, as probabilities of events are affected not only by the outcomes close by in space, but also those that have occurred recently in time. Ultimately, models such as those described above can be used as a framework for a statistical prediction system. Thus, the main aim of the chapter is to investigate space and space-time modelling and analysis procedures, applying firstly exploratory and then calibratory methods to the data on reported household burglary incidence in chapter 2 in the hope of deriving a specific space-time model, (which may or may not be based on existing models) to describe crime patterns. As yet, no work on making models of this type to explain crime patterns has been carried out, so a fairly comprehensive investigation will prove necessary.

4.2 Exploratory Examination of the Data

One of the initial purposes of examining the household burglary data is to identify and measure any patterns in time or space that may occur. Initially, space is to be considered in isolation. Maps of the distribution of crime incidence are first examined, and then various statistical tests are performed to shed some light on possible explanation of the spatial variations within these maps, and on the probabilistic processes which may be used to model geographical crime data such as this. After this, the time dimension is analysed, in order to discover seasonal, and temporally correlated

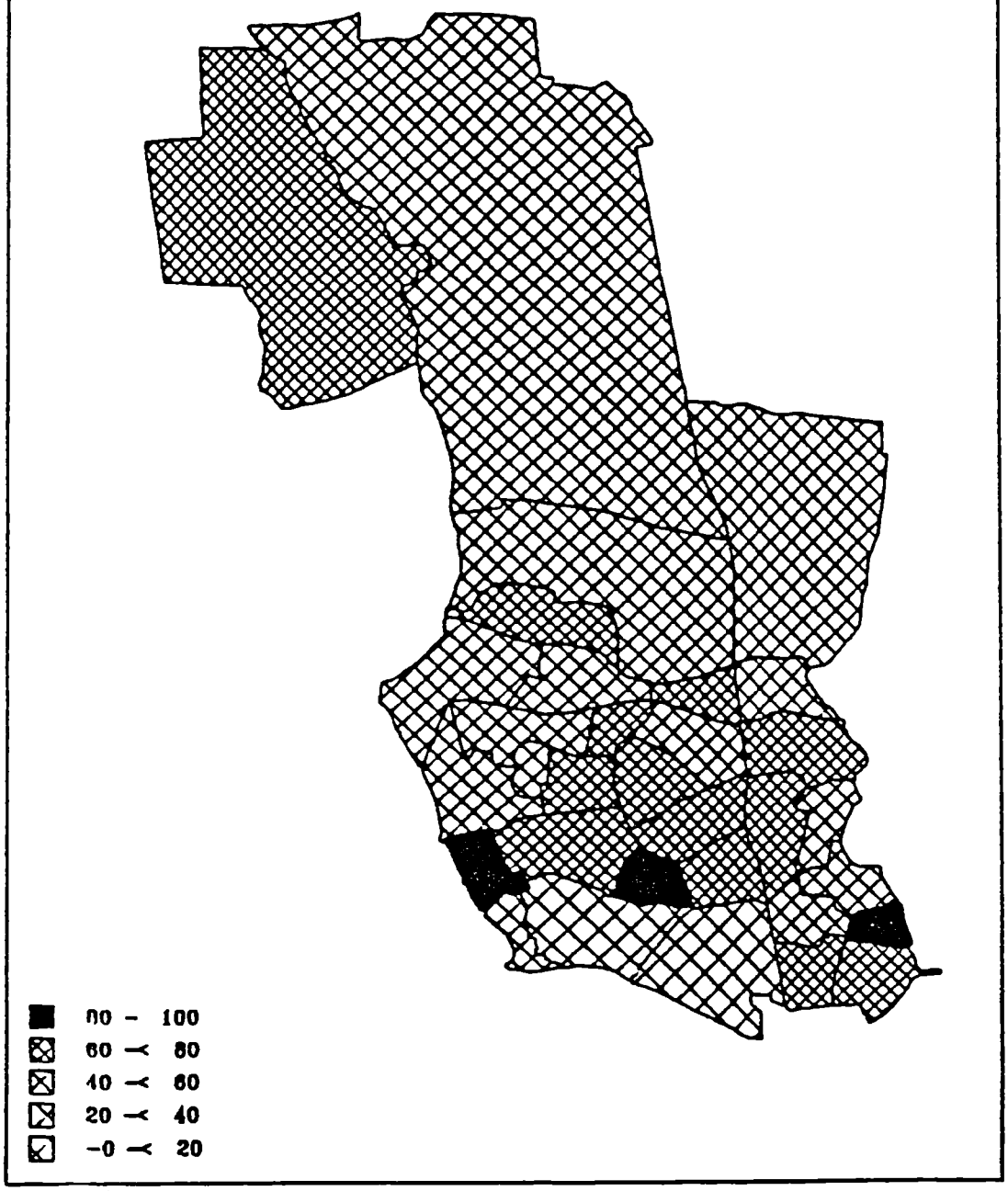
behaviour exhibited by the process. Finally both space and time are examined together, to see whether spatial processes are independent of time, or whether the process is one in which time and space interact. The tests and techniques used will be described in detail in each section.

4.2.1 The Spatial Distribution Of Burglary

In this section, various techniques are used to analyse the crime incidence in a purely geographical, or spatial sense. Initially, data for a single year will be considered. The spatial information information in the data set is specified to two differing levels of resolution. Firstly, and to the greatest degree of accuracy, there is a hundred metre total national grid square reference, based on the centroid of the postcode unit for the address at which the crime was recorded. Secondly, at a larger scale of aggregation, the code for the police foot beat region is also recorded. The latter could be deduced from the former, if point-in-polygon (Aldred, 1971) searching techniques were applied, but this could be time consuming computationally if it were to be done at every beatwise analysis, so it is better, given that the storage overheads are not unreasonable, to store both items of data. Although postcodes are in fact aggregate regions, rather than points in space, the level of resolution that they offer within the area of the subdivision under examination allows them to be treated as such. There are 1200 postcodes within the subdivision, but only 32 beats. An important implication of this is that point techniques designed for point spatial data may be

applied here, although at times it may be important to consider the aggregate nature of the data, for example when several events occur on the same street, giving the appearance of occurrence at identical points in space. The crimes reported over the first year are mapped in figure 4.1. Examination of this map suggests that there is marked heterogeneity within the spatial distribution of household burglary incidence, with certain areas being relatively free of crime, while others are "black spots". This observation in itself may not be regarded as particularly informative, from the viewpoint of analysing the process. It may only be of value if considered in conjunction with other geographically varying factors. For example, nothing has yet been said about the variation in housing density over the subdivision. It may be that these "black spots" simply correspond to areas of dense housing, but apart from allowing for the "population at risk" in this way, risk of burglary does not notably differ from house to house. Also, risk may be related to various social and economic variables which can be spatially referenced within the subdivisional area. However, although absolute figures in themselves may not be particularly helpful in this kind of analysis, they are of relevance to the police force resource management, since it is actual numbers of reported crimes that are most directly related to manpower demands, and maps of the spatial distribution of household burglaries that illustrate which areas require most attention. Also, a certain amount of analysis of the first type, in terms of burglaries per unit risk, may be carried out informally by police officers inspecting maps such as figure 4.1, who

Figure4.1: Household Burglaries by foot beat



view the area in the light of social, environmental and other geographical knowledge gained from working in the area.

4.2.2 Point Process Estimation: Techniques

Given the points considered above, it seems reasonable to consider the data both as a point pattern in its own right, and also in relation to other variables. As a point process, it may be useful to estimate a spatial probability distribution of the chance that a burglary occurs at each point within the subdivision from the year long point estimate crime sample. This may be visualised as a surface in three dimensional space, with x and y directions being used to cover the local geographical region, and the z direction being used to represent probability density. This could yield two useful statistics. i) Identification of regions where the probability density exceeds a certain level (high risk regions). ii) Beatwise averaged risks (obtained by integrating the surface over the beats in question). In order to do this, the method of Kernel estimation is applied (See eg Silverman, 1978a or 1978b). This is basically a technique used for obtaining an estimate of the probability density function from a set of point realisations of some process. To obtain the estimate, an expression of the form

$$f(h) = \frac{1}{kn} \sum_i g\left(\frac{x_i - h}{k}\right)$$

is used, where k is a smoothing constant, n is the number of points observed, and g is itself a probability density function, which is

normally symmetric. The effect of g is to create a "bump" centred on each point where an observation occurs, locally smoothing the distribution function estimate. Note that as g is a probability density function, then so is f. Various methods have been put forward for choosing k, but one of the best established is by informal trials at various values. Too low a value for k tends to result in under-smoothing, so that the estimate appears spikey, whilst too large a value tends to oversmooth, towards uniformity over the entire range of observed values. Note that although the equations shown here refer to the one-dimensional case, the same procedure may be extended logically to several dimensions. The two dimensional case gives

$$f(r,s) = \frac{1}{nk_1k_2} \sum_i \left(\frac{x_i - r}{k_1}, \frac{y_i - s}{k_2} \right)$$

The multidimensional extension to the kernel function g usually has symmetry in all directions (isotropy) and has a single mode at zero. Often, also, k1=k2. This seems reasonable as it assumes that if an observed event occurs at some point P in space, although it gives some information that points are likely to have these events occur near to it, it does not suggest anything about which directions in the locality are most at risk. Other methods of distribution estimation also exist. Some of these resemble the kernel method, perhaps allowing k to vary over the sampling space. Others base density estimates at a point on the distance to its jth nearest neighbour in the set of observed values (Discussed in Silverman 1978a). Distance is

assumed to vary inversely with density. However estimates of this type do not return true probability density function estimates, since as x tends to infinity, the density will vary as x^{-1}, and the integral of the density will not converge. Thus, estimates of this type, although of some use in examining behaviour in the locality of certain points, are not of use in estimating global density functions which will be required for mapping purposes. Also, this method requires ranking the observed value set from each point where an estimate is required by distasnce. This could be computationally expensive, particularly in two or more dimensions, where a sort would be required for every point at which an estimate was required. Methods using bandwidth variation could also be used, but again, for the purposes of exploratory examination, may prove computationally expensive. Thus, initially, a simple kernel estimator will be used for this study. Some choice should now be made for the form that g takes. One possibility is to take the normal distribution function,

$$g(\underline{x}) = \frac{1}{\sqrt{2\pi}} e^{-|\underline{x}|^2/2}$$

This is not without problems. If a function for g is chosen which will not become uniformly zero beyond a certain x-value, then for all observed x-values every single point at which f(x) is to be

estimated will be affected. If for example, the probability density function is to be evaluated on some grid with m elements, then nm evaluations of g would be necessary. However when some grid point is not particularly close to an observed point, g will be almost zero. Thus, several evaluations of g will have virtually no effect on the estimate. Thus it seems reasonable to choose some g having a finite nonzero domain, that is, having some value a for which $\text{mod}(\underline{x}) > a$ implies $g(\underline{x}) = 0$. It is also reasonable that g should lead to a smooth estimate, possibly no other reason than that most 3d graphics packages that could be used to represent this data will not handle discontinuities in any predictable or sensible way. Thus, cylindrical functions, such as

$$g(x) = 1/(\pi r^2) \qquad |x| < r$$

$$= 0 \qquad |x| \geq r$$

should also be avoided. If this is not the case, computations of high risk regions could also prove difficult, as these are most likely based on contouring methods, which like the 3d packages discussed above also work well only with continuous data. Therefore a continuous two-dimensional distribution is required, having finite nonzero domain. Epanechnikov (1969) has shown that functions of the form

$$g(\underline{x}) \propto 1 - \frac{|\underline{x}|^2}{r^2} \qquad |\underline{x}| < r$$

$$= 0 \qquad |\underline{x}| \geq r$$

with optimal bandwidth, approximately minimises the mean square error between the estimate of the probability density function and the true probability density function over the possibility space of $\underline{x}$. This quantity may be used as a measure of "goodness of approximation". This form of g, then, provides the least biased Kernel estimation surface (in an overall sense), if the bandwidth is well chosen. Bearing this in mind, an investigation using the dataset described in chapter 2 and kernel estimation using g as set out above will be carried out. A FORTRAN77 program is shown in listing 4.1. The program reads in points from the data set, and a bandwidth from the terminal input channel. From these it generates a kernel estimate of the probability surface over a square whose corners are given by the 8-digit national grid references 41805650 to 43005770, using the Epanechnikov kernels. The output of this will be a regularly spaced grid of density estimates, which may be fed as input to a contouring or surface drawing package, allowing visual representation of the surface.

4.2.3 Point Process Estimation: Results

The results of the kernel estimation using various bandwidths are given in figures 4.2 - 4.4. These surfaces were generated using the UNIRAS mapping and graphics software. As suggested earlier, lower bandwidth values give estimates that are sensitive to individual point values, whilst at very high bandwidths oversmoothing occurs, and the estimate takes on the shape of the kernel function with a high variance, with all observations when viewed on this scale appearing

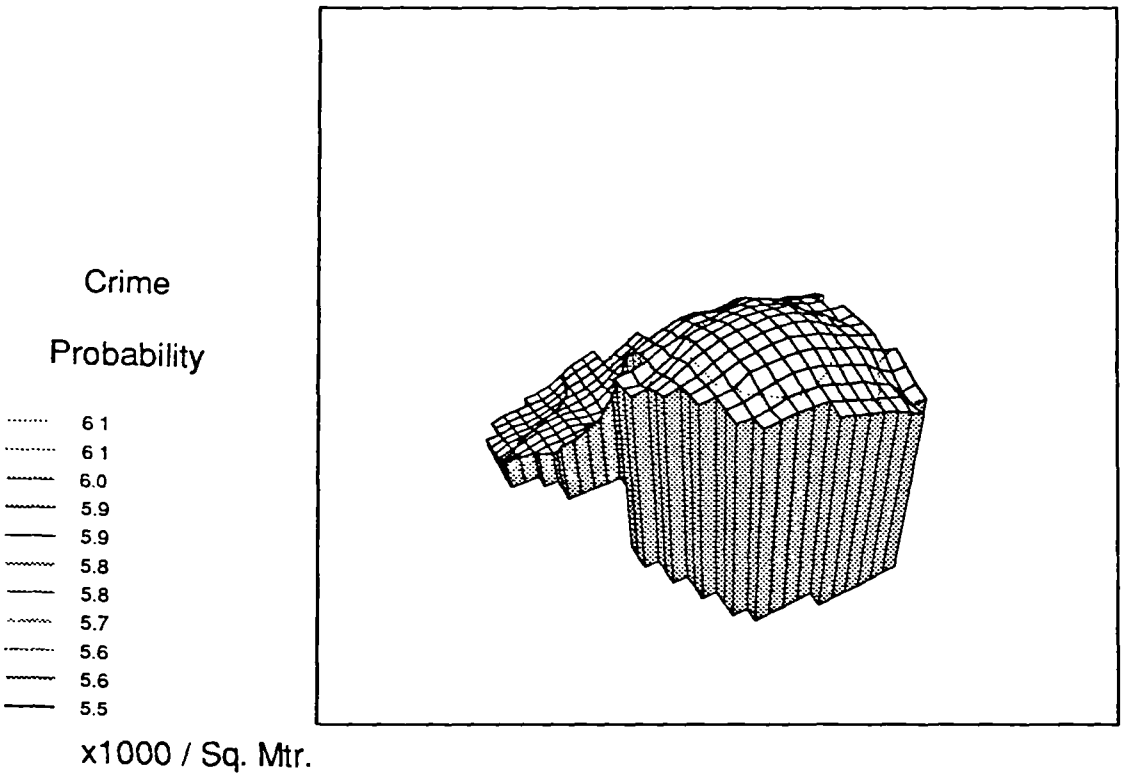


Figure 4.2

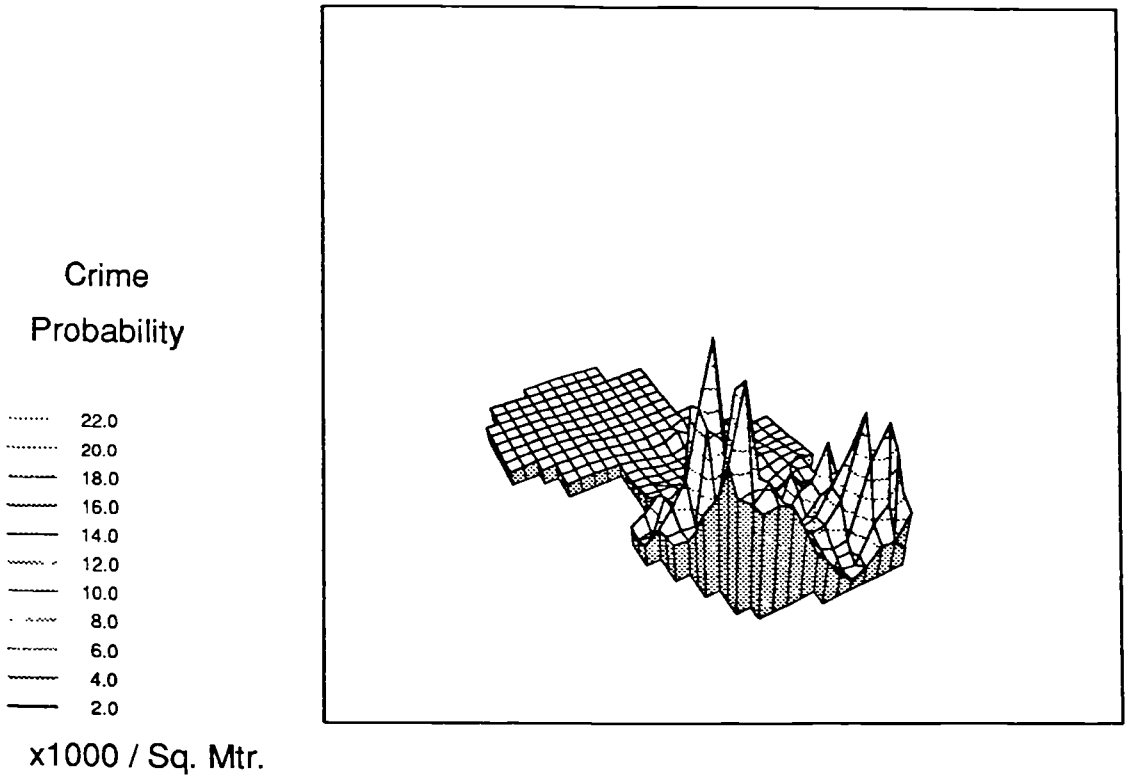

Crime
Probability
22.0
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
x1000 / Sq. Mtr.

Figure 4.3

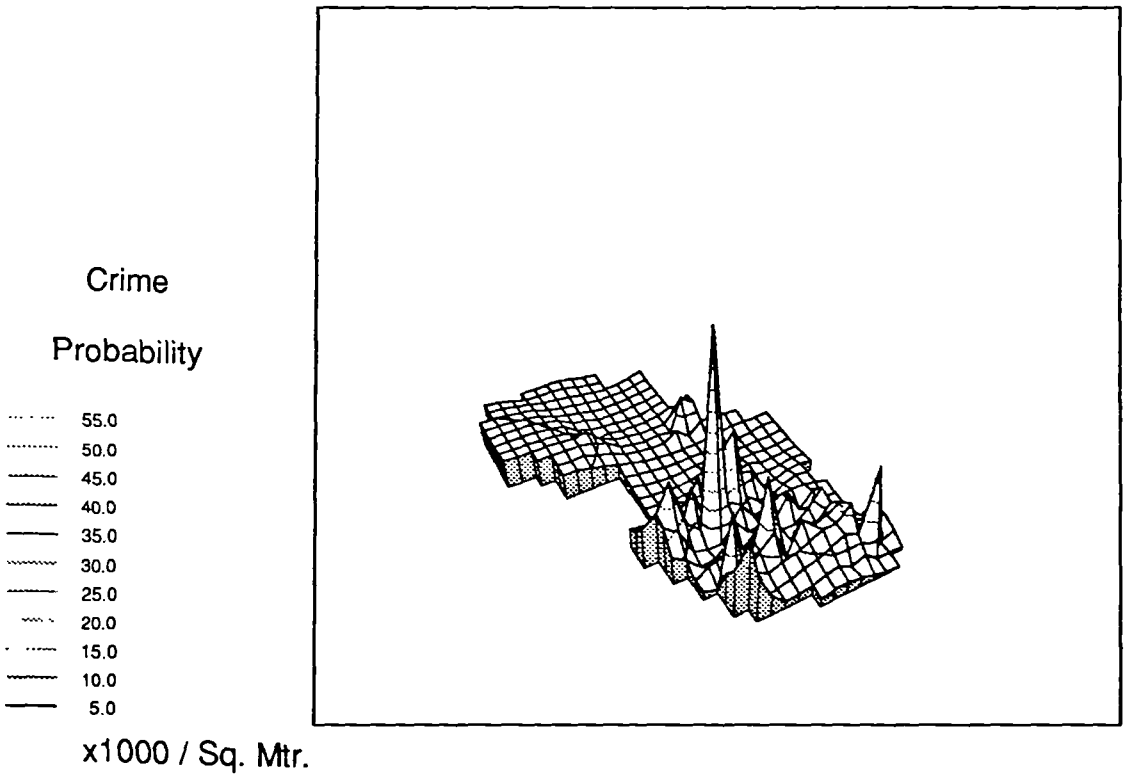

Crime
Probability
55.0
50.0
45.0
40.0
35.0
30.0
25.0
20.0
15.0
10.0
5.0
x1000 / Sq. Mtr.

Figure 4.4

clumped near the central mode. This type of behaviour is usual with kernel estimation. The most useful descriptive information is found from intermediate bandwidths. From such bandwidths it may be seen that the highest risk regions are those to the south of the subdivision, close to the edge of the central urban region. In the northern regions of the beat, where housing is less dense, virtually no burglaries occur. Discussion with police officers working in this subdivision supports this result. A further offshoot of discussion with these officers, when showing them various graphical representations of the spatial distributions of crime in the subdivision, is that the surfaces are capable of conveying more visual impact, and information, than either scatter diagrams, or three dimensional histograms or block diagrams, when examining data such as this for general spatial trends. For example, there are local modes in the density estimate, which could be identified by either a surface plot or contour diagram, and then referenced to the local geography if a map of relevant local features is superimposed. This benefit perhaps justifies this sometimes lengthy and strongly mathematical approach to spatial analysis as an alternative to simply producing scatterplots or histograms. Further quantitative analysis may be obtained from the kernel estimates by not only attempting to identify the modes of the distribution, but also those regions where the probability exceeds some chosen level. These may be thought of as "High Risk Zones". Of course, the boundaries of such zones can only be approximate, firstly because the kernel estimates themselves do not exactly specify the distribution (a finite set of points of data could not entirely specify a surface

function, if nothing at all is known about its functional form) and secondly because there is no objective and numerically precise definition of the boundary probability between "high risk" and "low risk". However, some arbitrary cut-point can be put set down, yielding a result that when mapped provides some useful insight into local geographic crime patterns at the exploratory stage of analysis.

Such a map is shown in figure 4.5. Here the value delineating the high risk zones from elsewhere is P=0.975. This value is chosen because the volume contained within the contour and the surface is approximately 5% of the total volume integrated over the subdivision. This calculation is demonstrated in appendix 1. This is equivalent to a 5% upper tail region in two dimensions, defined so that the risk of burglary at any point within the region exceeds that of any point outside of it.

In figure 4.6, the beat boundaries are shown superimposed over the high risk areas. Clearly, there is an overlap between the beats of the high risk areas. Thus, to some extent, it seems likely that any risk factors assigned to whole beats, in the form of some aggregated statistic, will exhibited noticeable spatial autocorrelation.

4.2.4 Housing Density

A very important factor, and one not yet investigated, is that of housing density. A useful though informal investigation of the relationship between housing density and risk of burglary is proposed

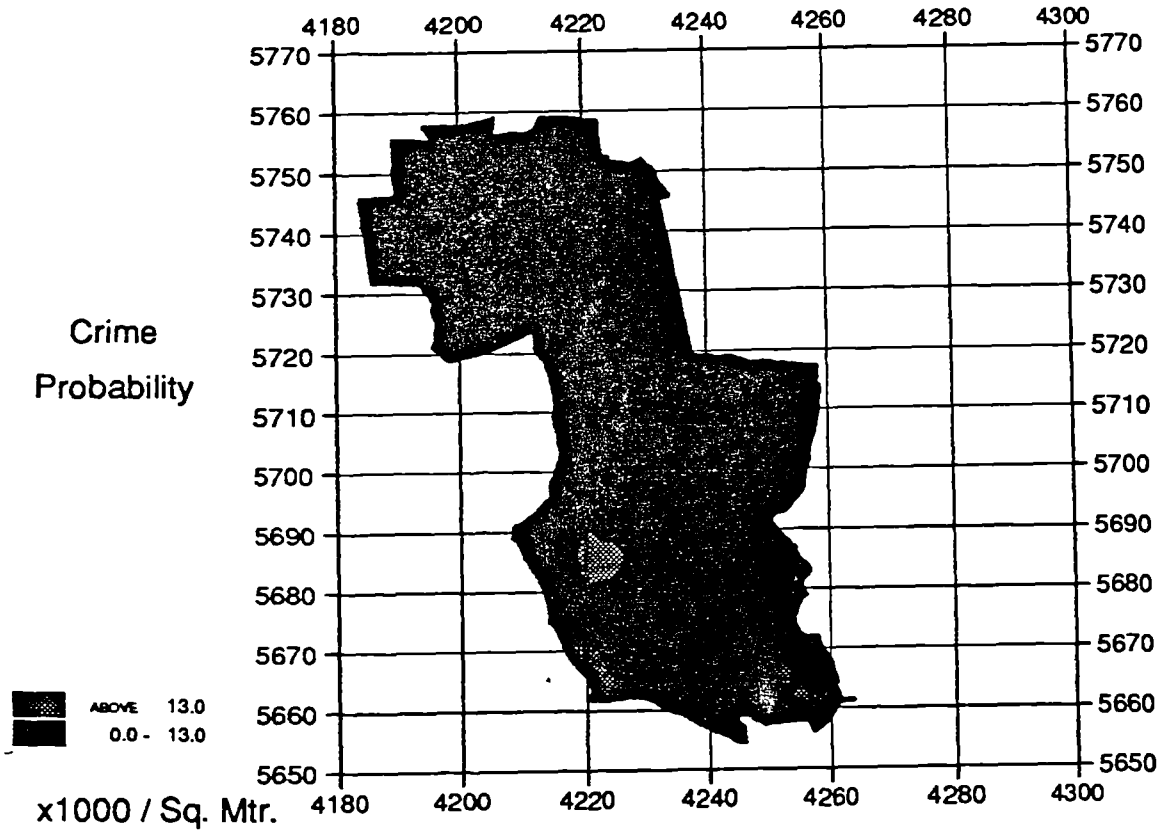

Crime
Probability
ABOVE 13.0
0.0 - 13.0
x1000 / Sq. Mtr.
4180
4200
4220
4240
4260
4280
4300
5770
5760
5750
5740
5730
5720
5710
5700
5690
5680
5670
5660
5650

Figure 4.5

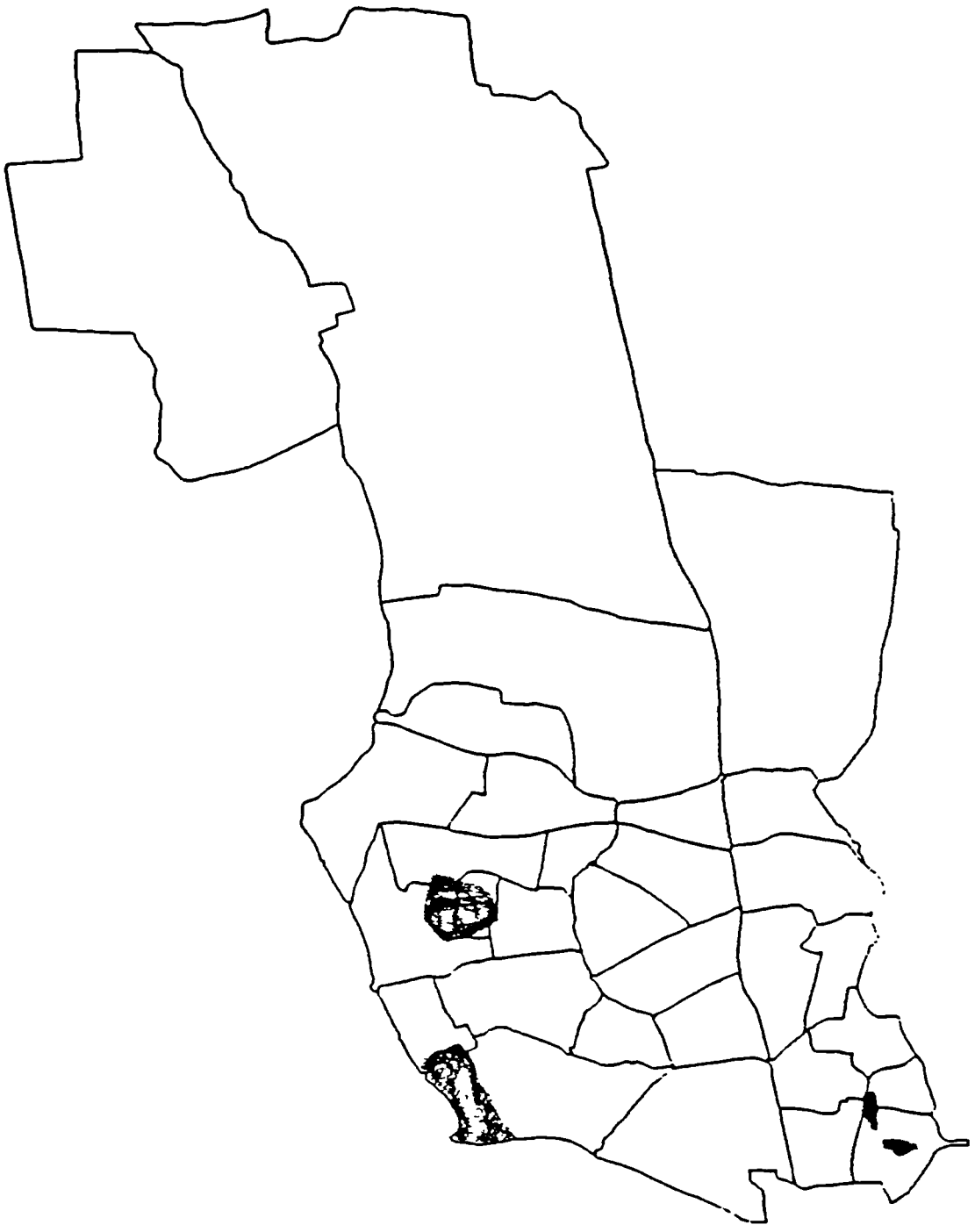


Figure 4.6

here. Using the mapping and graphics package UNIRAS again, a smoothed surface model of housing density may be created from a set of (x,y,z) triplets. Here x and y are the easting and northing of 1981 census EDs and z is the household count for the corresponding ED. These triplets are interpolated using a routine supplied with the package, allowing an estimated grid-based model of housing density to be fed into a 3-D graphics algorithm. Further to this, contours for crime risk may be superimposed upon this surface, allowing a visual comparison. If housing density alone can explain expected household burglary risk, one would expect to see similarly shaped contour line patterns, with high risk of crime corresponding to high density housing. An inherent problem with this approach is the reliability of the data. Since the most recent census data applies to 1981, it is possible that housing patterns may have changed to some extent between the time of the census and the time of the crime data survey. This is not too great a problem, as in a local area if there are notable discrepancies between housing densities and risks, it may be easy to check if housing has been developed or demolished in the intervening period in the areas in question. However, were this method to be carried into an automated system, perhaps over several subdivisions, such subjective checks would not be possible. This is a major caveat to adoption of a wider range of census variables on the prototype system.

An alternative graphical method can also be used. If the grid estimates for both the housing and burglary variables as densities are on grids of the same dimensions, a third variable may be created,

Household
Density
Projected onto
Crime
Probability

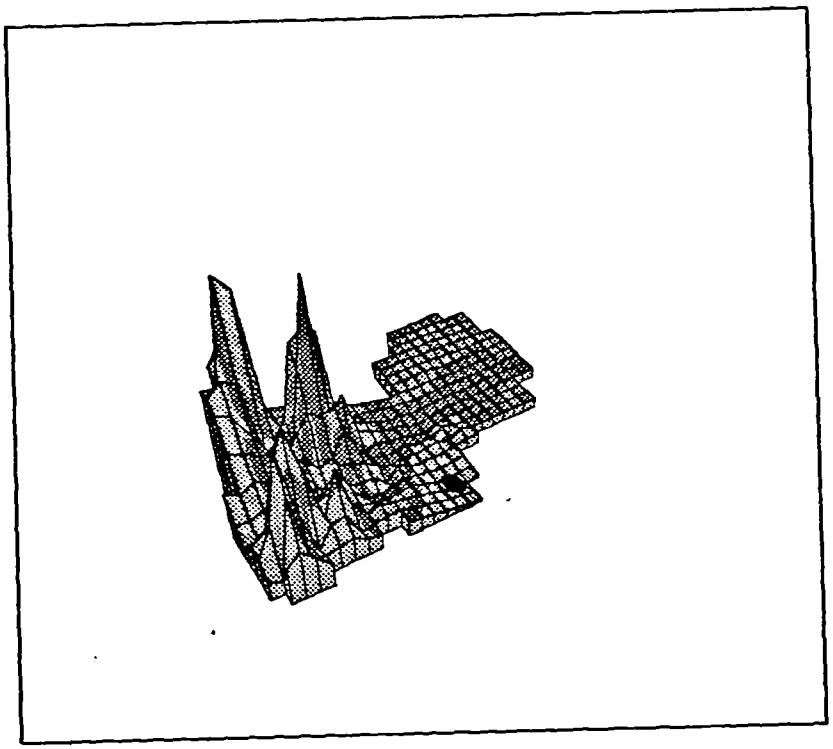


Figure 4.7

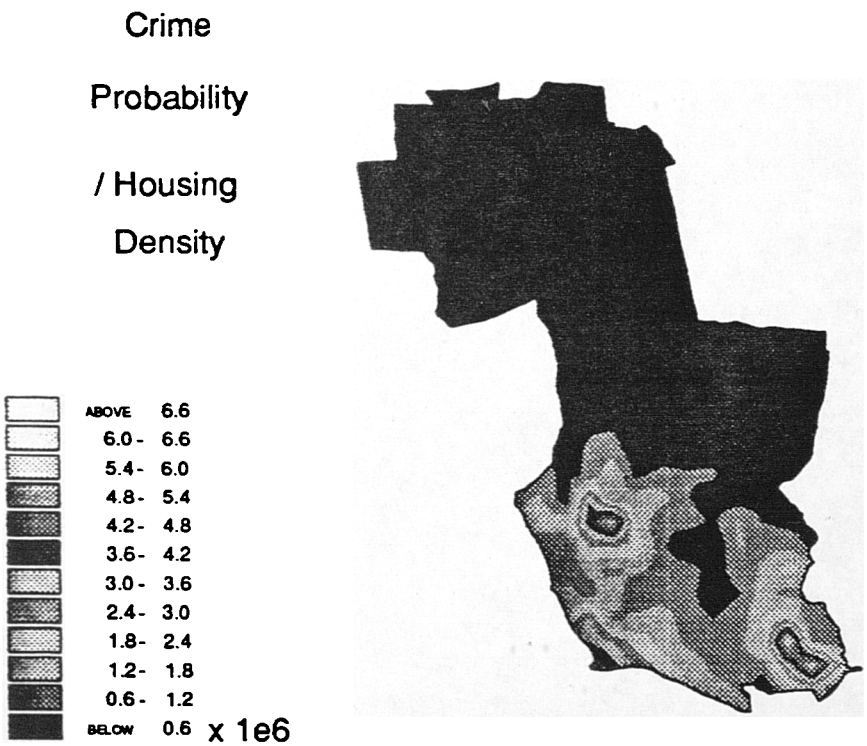

Crime
Probability
/ Housing
Density
ABOVE 6.6
6.0 - 6.6
5.4 - 6.0
4.8 - 5.4
4.2 - 4.8
3.6 - 4.2
3.0 - 3.6
2.4 - 3.0
1.8 - 2.4
1.2 - 1.8
0.6 - 1.2
BELOW 0.6
x 1e6

Figure 4.8

by computing the ratio of the first variable to the second. If burglary risk is proportional to housing density, one should expect a reasonably flat surface for this variable, as it varies over space. Both of these methods are illustrated in figures 4.7 and 4.8 respectively. It may be seen that, although there are areas peaking in absolute crime frequency also have dense housing, other areas of similar housing density do not exhibit such high burglary rates. These appear as "pot holes" on the second format of 3-D display. It appears that although housing density does contribute in some way to the likelihood of burglaries occurring in certain regions, it cannot be the only relevant factor. This is supported by the observed changeability of household burglary risk between regions of similar housing density in the study area.

4.2.5 Aggregate Spatial Analysis: Techniques

Having considered burglary as a point probabilistic process, where events appear as random points within two-dimensional space, the process will now be considered as being counts of crimes associated with the different police foot beat areas. It is necessary to understand the aggregated processes when predictions are to be carried out, since forecasting techniques capable of predicting exact points where crimes will occur do not exist. On the other hand, aggregated counts or rates of burglaries could be treated as a time or space-time series, and analysed in this framework for prediction purposes, since techniques for this type of data have been reasonably well established.

An initial analysis can be done by modelling the crime counts as a Poisson process. Firstly, to allow for seasonal variation, yearly crime counts will be considered. In the collected data, as well as individually coded events for the first year, there exist cross tabulated counts of crimes per beat per week for the following three years. As an initial overview, it may be worth considering the yearly crime rate as a Poisson process with a hazard function (Kalbfleisch and Prentice 1980) $\lambda(t)$, where lamda is a function of period one year.

Thus, the probability of a crime occurring in the interval {t, t+d} is if t is calibrated in yearly units. Thus, for each yearly count, the distribution will be poisson(k) where

$$K = e^{-\int_0^t \lambda(t)}$$

If K_t independent of past K.

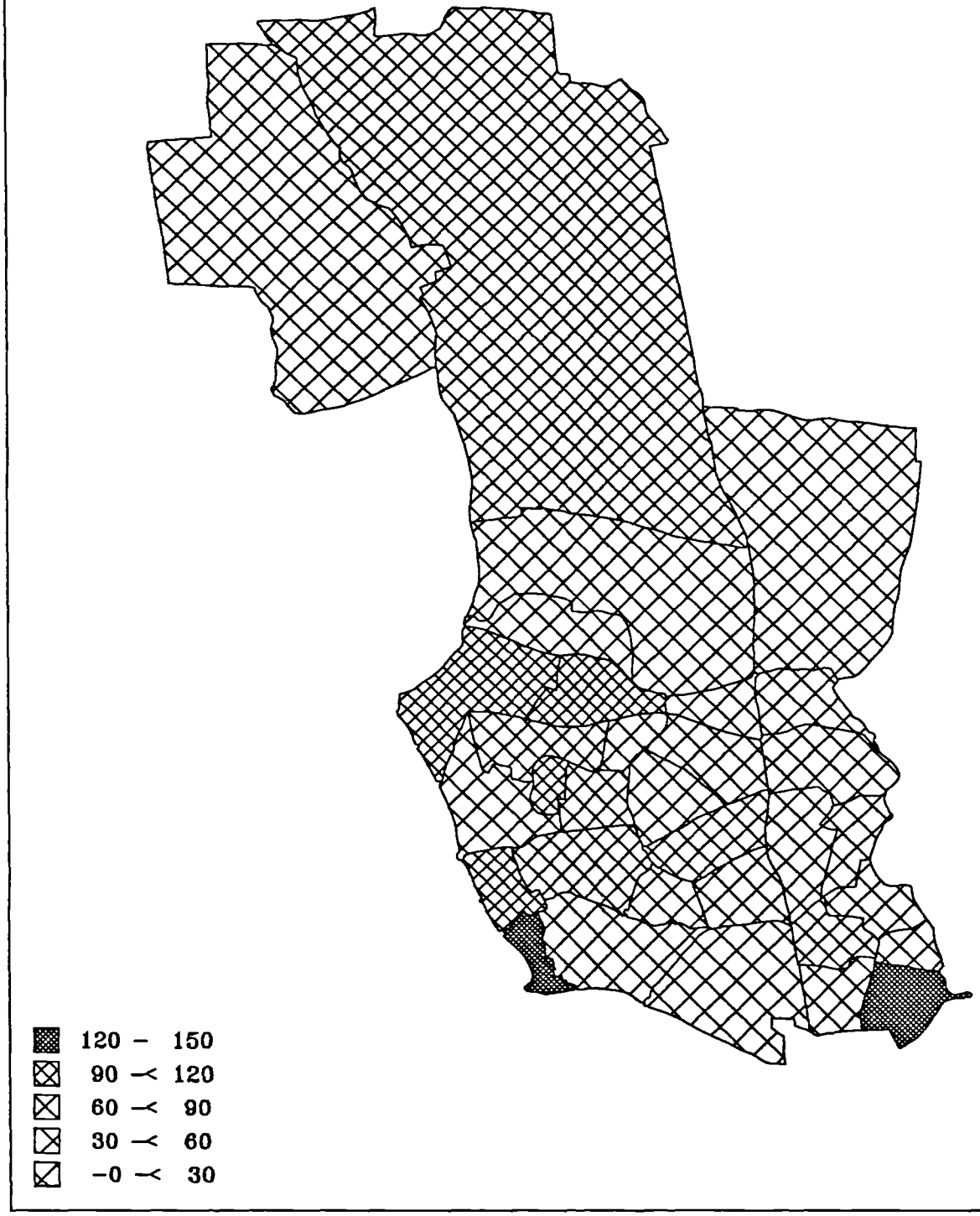
and the three yearly observations will be distributed independently. The maximum likelihood estimator of k will be the mean value over the three years for each beat. For each beat k is tabulated in table 4.1 and mapped in choropleth format in figure 4.9. It may be seen that the three high risk areas to the south of the subdivision are still highlighted, event after spatial aggregation to the beat level. These mean values can then be regressed against various social, economic and demographic variables estimated from the

Table 4.1

Yearly Burglary Rates By Police Beat

Beat	Average Rate
T1	73.67
T2	35.67
T3	38.33
T4	33.00
T5	51.33
T6	34.67
U1	42.00
U2	49.33
U3	127.67
U4	74.33
U5	53.66
U6	1.00
V1	47.33
V2	52.33
V3	57.67
V4	31.00
W1	37.67
W2	41.00
W3	71.67
W4	41.00
X1	30.67
X2	119.00
X3	105.33
Y1	34.00
Y2	77.00
Y3	108.00
Y4	87.67
Y5	53.33
Z1	75.67
Z2	124.33
Z3	94.67
Z4	2.00

Figure 4.9: Household Burglaries (Yearly Mean)



1981 census, if some investigation into geographical patterns link with household burglary incidence in the long term is sought.

If the model is to be applied strictly, then the regressions should be based on Poisson likelihood functions, as specified by a Poisson linear model of the form

$$\text{Burglaries} \sim \text{Poisson}(\mu)$$
$$\text{where} \quad \mu = \phi\left(\sum_{i=1}^{n} \alpha_i x_i\right)$$
$$x_i = \text{explanatory variables.}$$

Often the function phi is chosen to be the log function (See eg Bishop et al, 1975). However, in this case, the counts of observations are perhaps best converted into crime per household figures, and it may be more appropriate to apply the transformation $\sqrt{n+3/8}$ to the count as a dependent variable, which which will then become approximately Normal, with a fixed variance of 0.5 (Andscomb, 1948). If significance testing of the linear predictor variables is to be carried out, it must be borne in mind that each mean is based on three separate observations, supposed at this stage in the analysis to be independent, so that each observation is weighted three times. This cancels out any inaccuracy of significance testing or confidence limit estimation that may have been caused by assuming a sample of size n (where n is the number of beats) against the true value 3n.

Some important data that is needed in order to compute beatwise household burglary density is the household counts within each beat. Unfortunately these cannot be obtained exactly, since the smallest level of spatial resolution that such information available is the Census Enumeration District, and the boundaries of these zones do not necessarily coincide with those of the beats. However the counts may be approximated in the following way.

The centroid of each ED is available (although it is not clearly defined as to how it is computed), so that a simple estimator may be achieved by assigning each ED to the police foot beat containing its centroid using a point-in-polygon technique, and assigning its count of households to that beat. Summing over all of the EDs in a given beat will lead to the estimated household count for that beat. As long as the areas of the EDs are smaller than those of the beats by a reasonable amount, errors should not be too great since several EDs will be wholly contained in a foot beat. It is only those overlaying beat boundaries that could contribute to error.

This technique can also be applied to other census count variables that are tabulated to the ED level of spatial resolution, allowing regression modelling as an attempt to discover which aggregate beat characteristics best predict long run crime rates in those beats. Again at this point attention should be drawn to the problems raised earlier in the chapter when attempting this type of analysis. Firstly, there is a time lag between the census variables and the crime figures. In addition to this, measurement is being made

at an aggregate level, so the analysis is subject to the Modifiable Areal Unit Problem (Openshaw, 1984), and if the spatial patterns in the variables do not coincide well with the beat delineations, their effect may go unnoticed.

Notwithstanding all of these difficulties, there is still some value in carrying out the analysis. Certain patterns may become apparent, if sizes of fluctuations in space are sufficiently large scale to be detected in a set of beatwise aggregations. Also it may throw some light on characteristics of local geodemographics likely to influence crime incidence, and perhaps give certain clues about any spatial interdependence that might need to be incorporated into a stochastic spatial crime prediction model. For example, if the presence in a region of certain age ranges tends to correlate to higher crime rates, might this not suggest that neighbouring areas to those having large populations in this range are also at risk, if journey-to-crime routes cross the boundaries of adjacent beats?

Prediction of this type would not be feasible for a final model, however, since the independent variables could not be measured easily on a week by week basis, which would be necessary in a real-time short-term crime forecasting situation. In conclusion then, while this is reasonable at the exploratory analysis stage in order to provide ideas for a working model, the multivariate approach would not translate directly to such a system, due to the problems encountered in monitoring all of these variables on a weekly basis.

Up to this point, the analyses in this section relating to aggregate data methodologies have not allowed for spatial autocorrelation in the response variable. For example the previous regression based methodology although informally acknowledging that the processes are spatially linked, does not allow for this in the formal mathematical model. In fact, each beat is treated as an independently distributed variable, completely uncorrelated to its adjacent beats. This is justifiable to some extent, since the independent variables used to predict crime rates are also likely to be autocorrelated, and it is hoped that this will explain the autocorrelation in the response variable to some degree. However, given all of the pitfalls discussed above, and also given that the final prediction model will be required to predict future values for beats solely on the basis of past crime data, it will be of greater value to investigate the spatial correlation of crime counts, viewed as a realisation of a spatially interdependent probabilistic process.

This may be done in a number of ways. Two commonly used models for spatially autocorrelated processes, due to Moran (1950) are

$$E(Z_i \mid Z_j, j \neq i) = \mu_i + \sum_j c_{ij}(Z_j - \mu_j)$$

$$Var(Z_i \mid Z_j, j \neq i) = \sigma^2$$

(Conditional Autoregressive Model)

Where Z_i = crime count at beat i,

C_{ij} = matrix of similarities between beats

alternatively, one could use

$$Z_i = \mu_i + \Sigma C_{ij} (Z_j - \mu_j) + \xi_i$$

$$\xi_i \sim N(0, \sigma) \quad \text{independently.}$$

(Simultaneous Autoregressive Model)

Both of these models can be used to express equivalent processes (see Ripley, 1979). In each case the error is normally distributed with mean zero and variance σ^2. This is not unreasonable if the crime counts are poisson distributed, and thus $\sqrt{n+3/8}$ is approximately normal. Moran (1950) proposed a measure of spatial autocorrelation to be

$$I = [n \Sigma w_{ij} (x_i - \bar{x}) (x_j - \bar{x})] / [(\Sigma w_{ij}) \Sigma (x_i - \bar{x})^2]$$

Similarly, Geary suggested

$$C = [(n-1) \Sigma w_{ij} (x_i - x_j)^2] / [(2 \Sigma w_{ij}) \Sigma (x_i - x_j)^2]$$

The coefficients of w were initially assumed to be 0/1 adjacency indicator variables, although Cliff and Ord (1973) proposed that these could be generalised to any matrix, as a continuous

similarity measure. The coefficients I and C can both be thought of as tests of $\rho = 0$ in the hypotheses

$$\underline{X} = \underline{\mu} + \rho W (\underline{X} - \underline{\mu}) + \underline{\varepsilon}$$
$$\underline{\varepsilon} \sim N(0, \sigma^2) \quad \text{independently}.$$

In this test W is assumed to be known, leading to some difficulty. Alternative possibilities for W could be

1) A simple dij matrix of contiguity
2) Some monotone decreasing function of distance between beats
3) Some monotone increasing function of common boundary distance of beats
4) Some measure of social similarity (a "distance" between aggregate census variables.)
5) Some combination of any of the above.

Consideration of this problem is given in Cliff and Ord (1973) and Hagget et al (1977). Bartels (1979) suggests that the simple contiguity matrix has proved to be as adequate as other, more sophisticated postulated matrix coefficient models. However, this may not be the case here, where there is a great deal of variation in the size of the foot beat areas. Thus, parts of the large north western beat (see eg. figure 4.9) are up to 10km from the nearest neighbouring beat, whilst in certain urban beats, no point within the boundaries is more than about 3km from the nearest beat. In the simple contiguity model, the crime rates for each pair of adjacent beats would be represented as being equally correlated, which is

unlikely to be the case, particularly if in the situation of rural beats where the centres of population are not near to the coincident borders.

Adopting a matrix based on common boundary lengths also seems problematic, again due to the variation in beat areas. Large rural beats like that to the north west of the study region have long boundaries in common with other neighbouring beats, which would imply strong correlation using this model. In addition, the smaller inner city beats would have relatively small correlations implied, whereas journey to crime discussions (Pyle, 1974 or Evans, 1980) would suggest that these beats, in zones of similar high housing density, are more likely to interact. It would seem that there is some danger of the common boundary criteria discriminating between likely and unlikely correlates in the opposite direction to that of a desirable target model.

This leaves distance based metrics, in purely space-time based systems, or possibly some combination in these and social, housing and economic measures. A reasonable purely distance-based measure might be the distance between the centres of population for each beat-pair. These could be estimated using ED-based counts: For each ED whose centroid lies within a given beat, if P(i) is the population of the ith ED within the beat, and $\underline{X}_i$ is its centroid (expressed as a vector) then the centre of population could be defined as

$$\frac{\sum_{i=1}^{n} P(i)\,\underline{X}_i}{\sum_{i=1} P(i)}$$

Clearly, one source of error here will occur when the EDs straddle beat edges. Thus, such a matrix will be subject to a sensitivity test, by reassigning edge-coincidence EDs to the neighbouring beat. Figure 4.10 shows the initial set of centroids, and figure 4.11 shows those for which some edge EDs have been reassigned to adjacent beats, after random selection with a 50% probability of reassignment. No large variation in centres of population has occurred. Thus it is perhaps not unreasonable to consider the weighting matrix using this method. In addition to this, it is also possible to consider a similar distance measure, based upon centroids weighted by household counts.

Given the problems with census variables due to time-lags and spatial registration, it may be as well not to use these as a basis for measurement in any great amount. Thus it would seem wise to consider mainly distance-based matrices. To this end, four possible matrices will be considered. Firstly the simple 0/1 contiguity matrix, then the housing and population centroid distance matrices, and finally one based on beat area centroids, defined as

$(\bar{x}, \bar{y})$ where $\bar{x}$ and $\bar{y}$ are centroids of x and y elements of a vector definition of beat outline

the simplest form of distance-based model. If the difference between the simple models and the more complex, although more carefully designed models is slight, considerable computation time may be saved by adopting these in the prediction system. However,

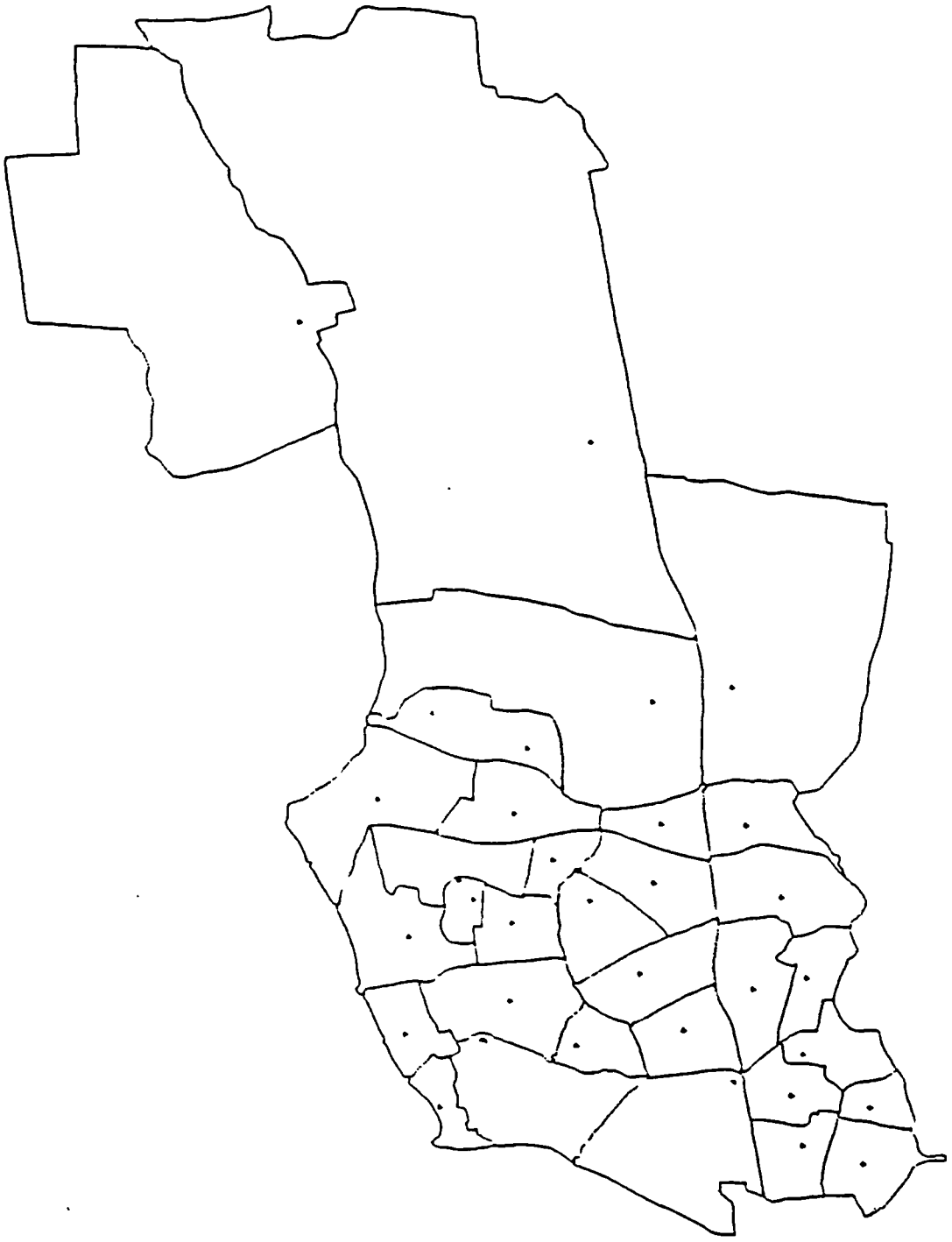


Figure 4.10

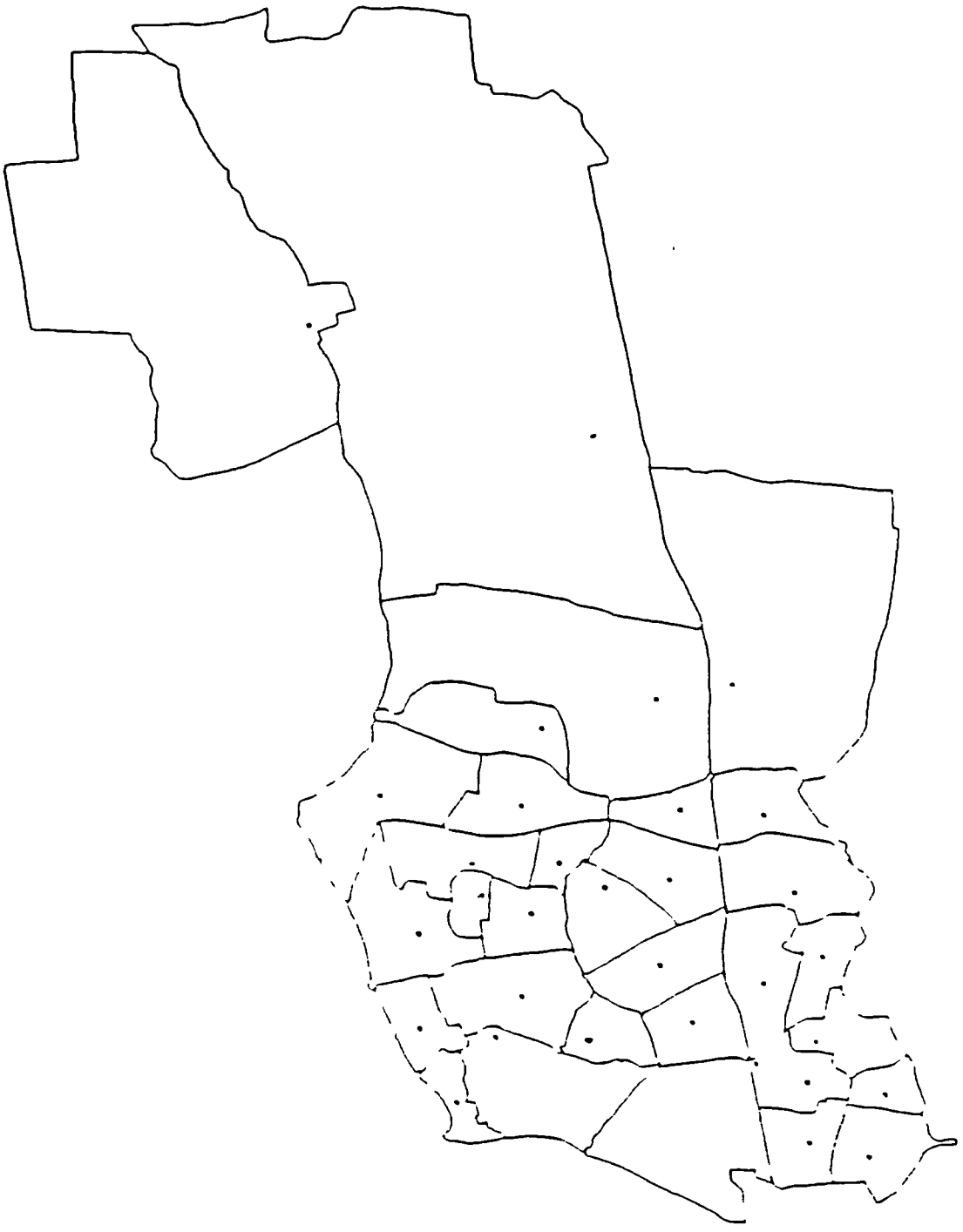


Figure 4.11

it is necessary to examine the more complex systems initially to see if this is in fact the case. The aim of this part of the analysis will then be to discover which, if any of the proposed matrices might be used to model a space-time series used to predict short term crime densities.

4.2.6. Aggregate Process Analysis : Results

Firstly, the regression model is considered. The variables to be incorporated in this model are listed in table 4.2. Certain variables are liable to be correlated, and so a crosstabulation of the Pearson correlation coefficient is given in table 4.3. In each cell in this table, the first quantity is the estimate of the coefficient, and the second, in brackets, in the significance level of this value against a null hypothesis of no correlation. The table suggests that, at least in terms of prediction, if not in those of substantively explaining the processes occurring, not all variables need be incorporated in the model since there are certain clear cases of high correlation between predictor variables. A stepwise or back-substitution technique could be used to eliminate superfluous variables.

Initially, the simple correlation between each variable and the crime rates are examined (table 4.4). Many of the census variables exhibit strong correlation with the crime rates.

Table 4.2
Proposed Independent Variables For
Crime Risk Model

Description	Variable Name	Census Derivation
Population	POPN	C937
No. Households	HOUSES	C929
Proportion "Young"	YOUNG	10000 * (C71 + C78 +C85 +C92 +C99) DIV C50
Propn. LA Housed	COUNCIL	10000 * C983 DIV C929
Propn. Male UE	MALEUN	10000 * C860 DIV C720
Propn. Youth UE	YOUNGUN	10000 * (C865 + C870 + C875) DIV (C725 + C730 + C735)
Bus Journey To Work	BUSJTW	10000 * C4731 DIV (C4411 + C4412)
Overcrowding	OVERCRWD	10000 * C945 DIV C929
3 Cars or more	THREECAR	10000 * C1174 DIV C1170
Owner Occupied	OWNOCC	10000 * C967 DIV C929
Furnished Rental	FURNRENT	10000 * C1063 DIV C929
Retired Persons	RETIRED	10000 * C1669 DIV C1629
Single Households	SINGLEH	10000 * C1360 DIV C1351

The C-codes refer to census variable names.

Table 4.3

Pearson Correlation Coefficients

	POPN	HOUSES	YOUNG	COUNCIL	MALEUN	YOUNGUN
POPN	1.00000	0.96290	0.22040	-0.00903	-0.09205	-0.29825
	0.0000	0.0001	0.2419	0.9622	0.6285	0.1094
HOUSES	0.96290	1.00000	0.20160	-0.11339	-0.13620	-0.34957
	0.0001	0.0000	0.2854	0.5508	0.4730	0.0583
YOUNG	0.22040	0.20160	1.00000	0.47135	0.66901	0.56029
	0.2419	0.2854	0.0000	0.0086	0.0001	0.0013
COUNCIL	-0.00903	-0.11339	0.47135	1.00000	0.81417	0.74935
	0.9622	0.5508	0.0086	0.0000	0.0001	0.0001
MALEUN	-0.09205	-0.13620	0.66901	0.81417	1.00000	0.94345
	0.6285	0.4730	0.0001	0.0001	0.0000	0.0001
YOUNGUN	-0.29825	-0.34957	0.56029	0.74935	0.94345	1.00000
	0.1094	0.0583	0.0013	0.0001	0.0001	0.0000
BUSJTW	0.14481	0.07943	0.56154	0.76656	0.84326	0.72321
	0.4452	0.6765	0.0013	0.0001	0.0001	0.0001
OVERCRWD	-0.05443	-0.10205	0.56754	0.64028	0.85867	0.79963
	0.7751	0.5915	0.0011	0.0001	0.0001	0.0001
THREECAR	-0.27651	-0.30032	-0.37732	-0.59468	-0.56442	-0.36465
	0.1391	0.1069	0.0398	0.0005	0.0012	0.0476
OWNOCC	0.06342	0.06035	-0.53585	-0.85114	-0.82839	-0.72846
	0.7392	0.7514	0.0023	0.0001	0.0001	0.0001
FURNRENT	0.00252	0.18549	0.23311	-0.51490	-0.17267	-0.16742
	0.9895	0.3264	0.2151	0.0036	0.3615	0.3765
RETIRED	-0.28463	-0.14350	-0.62178	-0.20030	-0.20710	-0.22690
	0.1274	0.4493	0.0002	0.2886	0.2722	0.2279
SINGLEH	-0.27478	-0.03732	0.15103	-0.15832	0.08752	0.06272
	0.1417	0.8448	0.4257	0.4034	0.6456	0.7419

Table 4.3 (continued)

	BUSJTW	OVERCRWD	THREECAR	OWNOCC	FURNRENT	RETIRED	SINGLEH
POPN	0.14481	-0.05443	-0.27651	0.06342	0.00252	-0.28463	-0.27478
		0.4452	0.7751	0.1391	0.7392	0.9895	0.1274
HOUSES	0.07943	-0.10205	-0.30032	0.06035	0.18549	-0.14350	-0.03732
		0.6765	0.5915	0.1069	0.7514	0.3264	0.4493
YOUNG	0.56154	0.56754	-0.37732	-0.53585	0.23311	-0.62178	0.15103
	0.0013	0.0011	0.0398	0.0023	0.2151	0.0002	0.4257
COUNCIL	0.76656	0.64028	-0.59468	-0.85114	-0.51490	-0.20030	-0.15832
	0.0001	0.0001	0.0005	0.0001	0.0036	0.2886	0.4034
MALEUN	0.84326	0.85867	-0.56442	-0.82839	-0.17267	-0.20710	0.08752
	0.0001	0.0001	0.0012	0.0001	0.3615	0.2722	0.6456
YOUNGUN	0.72321	0.79963	-0.36465	-0.72846	-0.16742	-0.22690	0.06272
	0.0001	0.0001	0.0476	0.0001	0.3765	0.2279	0.7419
BUSJTW	1.00000	0.75034	-0.71113	-0.75062	-0.21765	-0.15014	0.01238
	0.0000	0.0001	0.0001	0.0001	0.2479	0.4284	0.9482
OVERCRWD	0.75034	1.00000	-0.37816	-0.67931	-0.05669	-0.12345	0.01211
	0.0001	0.0000	0.0393	0.0001	0.7661	0.5157	0.9493
THREECAR	-0.71113	-0.37816	1.00000	0.65238	0.13942	-0.10687	-0.19220
	0.0001	0.0393	0.0000	0.0001	0.4625	0.5741	0.3089
OWNOCC	-0.75062	-0.67931	0.65238	1.00000	0.13746	-0.02820	-0.22877
	0.0001	0.0001	0.0001	0.0000	0.4689	0.8824	0.2240
FURNRENT	-0.21765	-0.05669	0.13942	0.13746	1.00000	0.05058	0.59348
	0.2479	0.7661	0.4625	0.4689	0.0000	0.7907	0.0006
RETIRED	-0.15014	-0.12345	-0.10687	-0.02820	0.05058	1.00000	0.33892
	0.4284	0.5157	0.5741	0.8824	0.7907	0.0000	0.0669
SINGLEH	0.01238	0.01211	-0.19220	-0.22877	0.59348	0.33892	1.00000
	0.9482	0.9493	0.3089	0.2240	0.0006	0.0669	0.0000

The upper figures refer to correlation coefficients, and the lower figures to their significance.

Table 4.4

Correlation Of Explanatory Variables With Crime Rates

Variable	Coefficient
POPN	0.53803
	0.0022
HOUSES	0.51794
	0.0034
YOUNG	0.74778
	0.0001
COUNCIL	0.37611
	0.0405
MALEUN	0.60580
	0.0004
YOUNGUN	0.46088
	0.0104
BUSJTW	0.59051
	0.0006
OVERCRWD	0.55223
	0.0016
THREECAR	-0.45124
	0.0123
OWNOCC	-0.41009
	0.0244
FURNRENT	0.16145
	0.3940
RETIRED	-0.50738
	0.0042
SINGLEH	0.05038
	0.7915

Lower figure represents significance of difference of coefficient from zero.

A stepwise regression model is now run. The purpose of this is not so much to see which variables are eventually to be included (Indeed, variation of the significance levels for entering and dropping variables can lead to variation in this final set of variables) but to discover what level of between beat variation can in fact be explained by a model of this sort.

The results are tabulated in table 4.5. It may be observed that 82% of variation may be accounted for in this manner. Variables relating to unemployment figure highly, as do models relating to demographic age profiles of the beats. It is hard to interpret whether these variables are reflecting characteristics making people likely to commit crimes, or likely to be victims, or some mixture of both of these effects. Note that the house crowding indicator is also a strong predictor. This is perhaps not surprising, as it may be a proxy for the types of housing characterising the area as a whole. Theories such as those of defensible space (Newman 1972, 1976) suggest that certain types of housing are at greater risk.

Concluding this analysis, it is important to recall that a final system cannot be expected to have all of these variables constantly monitored in any formal way, and interpretation of this must be done in terms of how one might expect a process involving only crime rates themselves to behave. One conclusion is that if certain age groups are more likely to commit crime than others, then given there may also be transport constraints limiting journey to crime distances, areas nearer to housing with high concentration of these age groups will be

Table 4.5

Results Of Stepwise Regression

STEP	VARIABLE ENTERED	VARIABLE REMOVED	NUMBER IN	PARTIAL R**2	MODEL R**2
1	YOUNG		1	0.5592	0.5592
2	POPN		2	0.1464	0.7056
3	YOUNGUN		3	0.0861	0.7917
4	COUNCIL		4	0.0331	0.8248

at greater risk. More generally, if position in space does effect risk of burglary, a certain degree of spatial autocorrelation is bound to occur in the observed rates. Further evidence supporting this suggestion can be based on the theories linking housing design and risk of crime. For example, in more modern housing schemes in which houses are similarly styled, possibly with security being taken into account in the design process, then a reasonably sized cluster of housing will have similar low risks of burglary. The converse may apply to an older area. Thus, one expects risk to be spatially correlated over regions of similar housing, from house to house, possibly street to street, and over entire estates. All of these arguments lead to the idea that the data for crime rates alone should exhibit spatial autocorrelation, which will now be investigated in its own right.

At this stage, it seems appropriate to examine the degree of spatial correlation between the beats, and to attempt to model the structure of this correlation. The results of this, unlike those above, may be directly incorporated into a model of a stochastic process that may be used directly in a system to predict the crime rates.

As discussed earlier in the section on techniques, various systems of autocorrelation may be modelled (based on the W-matrix), and tested as an alternative hypothesis to a system of

independently distributed zones, showing no spatial interaction. As put forward in the techniques discussion, four basic models will be used. Firstly a simple contiguity matrix, and then three distance matrices, for distances between housing centroids, population centroids and areal centroids (mean centres) for beat pairs. It is suspected that the similarity measure for the weighting matrix should be a monotone decreasing function of the distances. A reasonable model might be to have Wij = d_{ij}^{-1} where dij is the distance between beats i and j using one of the above definitions, and a is a positive real number. Initially, values for a of 1 and 2 will be considered. Some consideration will then be given to estimating a maximum likelihood estimate of a. However, the initial models will be fitted using a=1 or a=2.

For any of these test methods, the Morans-I coefficient may be used as a test of the hypothesis rho=0 in the equation above. The Geary's-C coefficient may also be used, but Cliff and Ord (1973) demonstrated that I is preferable to C in simulation studies, and by showing that the relative efficiency of I to C is always greater than or equal to unity. Note that, although an I-statistic may be computed for each different hypothesis (with respect to each of the W-matrices) significance tests will not be independent.

Tests were carried out for each of the matrices, using the recommended procedure by Cliff and Ord (1973) for approximating the upper tail of the distribution of the I-statistic. As usual the X-variates were the square root based transformations of the crime

density per household, due to the approximate normal distribution of this quantity under a Poisson crime count assumption. The results are listed in table 4.6. As can be seen, the only formulation for the W-matrix that did not prove significant is the contiguity based model. It could be that as beats do vary greatly in area, and in their internal population geography and housing geography, that simple contiguity is unhelpful in explaining interdependence. This is particularly likely if the effects of one or two large rural beats are weighted beyond their importance in their effect on adjacent, possibly equally large and remote beats.

It is apparent here that more significant results are obtained when a=2 than when a=1. It may be useful to find a better approximation of a, perhaps based on maximum likelihood estimation. For the CAR model the log-likelihood of the scheme is given by

$$-\frac{n}{2}\ln(2\pi\sigma^2) + \frac{1}{2}\ln|B| - \frac{1}{2\sigma^2}(\underline{z}-\underline{\mu})^T B(\underline{z}-\underline{\mu})$$

where

$$B = I - C$$

Table 4.6

Moran's I-Coefficient For Differing Spatial Weighting Matrices

Characteristics			Results	
Exponent	Correction for no. Houses	Distance Metric	Moran's I	Significance
-	Y	Contiguity 0/1	-0.0718	0.6122 NS
-	N	Contiguity 0/1	0.5730	0.2513 NS
1	Y	Household Centre	0.0576	0.0125 *
1	Y	Population Centre	0.0678	0.0063 **
2	Y	Household Centre	0.4441	0.0000 **
2	Y	Population Centre	0.5507	0.0000 **
1	N	Household Centre	0.0388	0.0482 *
1	N	Population Centre	0.0387	0.0525 NS
2	N	Household Centre	0.1498	0.0849 NS
2	N	Population Centre	0.1663	0.0925 NS

Note that the coefficient a would appear in several terms of the determinant of B, which could prove problematic. However, this may be overcome if Pseudo-likelihood estimation is carried out (Besag, 1975,1977). The Pseudo-likelihood is defined as

$$PL = \prod_i Pr(z_i \mid z_j, j \neq i)$$

which is to be maximised to estimate a, sigma and c. For a CAR process,

$$\ln(PL) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|(I-C)(\underline{z}-\underline{\mu})\|$$

Minimising this reduces to finding σ^2 as the mean square of the residuals $(\underline{z}-\hat{\underline{\mu}}) - \hat{C}(\underline{z}-\underline{\mu})$, and choosing $\hat{C}$ to minimise σ^2 (Ripley, 1979). However, this process has been found to be problematic, since it requires the repeated evaluation of a 32-dimensional determinant, in order to iteratively maximise the above expression. This, and possible difficulties with rounding error in a computation of this size, suggest that the integer trial values alone should be included in the study. These show reasonasble correlation in any case, if a reasonable choice of distance metric is used.

Thus, a reasonable estimate of a spatial process for crime rates is given by a CAR model with Wij = d_{ij}^{-2}. This may be used in simulations or calculations for predicting crime rates in this region.

4.3 Time Series Analysis

Having gained some understanding of the household burglary data when considered as a realisation of a spatial stochastic process, the next major step is to analyse the data in the time dimension. This is essential for short-term forecasting techniques could not be made without an understanding of how future crime rates are linked to those in the past, in some regular way. It is expected that some autocorrelation in time should occur, since informal conversation with police officers suggests that burglaries in particular areas occur in temporal clusters (ie offenders show a tendency to return to the same area and repeat offences over short lengths of time), and this may also be the case for other types of crime. It is also possible, in some instances, that this correlation may be negative. Consider, for example, a situation where a large number of burglaries in an area may lead to increased police presence, which may in turn then lead to a reduction in criminal activity. However, although this autocorrelation has been conjectured, it remains to examine the data for empirical evidence. Also, little is known about the length of time over which correlation might occur. This second fact is of great relevance, as it answers the question "how far back in time must past crime counts be stored in order to obtain reliable predictions of the coming weeks values?". The following section deals with the statistical methodology used to investigate these questions.

4.3.1 Methodology

Initially, the total number of crimes each week within the entire subdivision will be considered as a single time series, but eventually this should be expanded to allow for differing predictions to be made for different localities. The purpose of initially examining at this large scale is to obtain some idea of the time scales that the stochastic process work in, without the added complication of how this interacts within space. Some of the results in time at this scale may then be used as a point of departure for modelling the more complex space and time process in the final model building stage. A further defence of the aggregate analysis approach, at least initially, is that summed data of this type is more nearly Normally distributed, and although some non-parametric techniques exist for time series analysis, the main bulk of the subject lies in parametric models involving Normality. Eventually, of course, smaller scale beatwise systems must be considered, but early hypothesis tests, based on parametric Normal models, can be made at this larger scale.

A review of literature on time series analysis suggests (eg Glass et al, 1975) that two main streams of analysis procedure exist: That based on the frequency domain, and that based in the time domain. The former attempts to account for the variance of a variate evolving in time by partitioning it into components associated with oscillations at various frequencies, whilst the latter views a the same process by relating the current value of the variate to those observed in the past, and to other time-referenced random processes.

The second type of analysis seems more appropriate for the crime prediction application. Past weekly crime counts are easily extractable from the database, and these could then be used for predictive algorithms in the crime pattern analysis system.

In contrast to analysis in a spatial framework, well established methods of time series classification and identification already exist. One of the most common is that given by Box and Jenkins (1970), which has already had wide application in the field of economics and several other areas. It proposes a family of stochastic processes, which is sufficiently general to cover a diverse range of situations. For example, one of its members yields the model for a time series used to derive the exponential smoothing technique, but another gives rise to the naive technique. Many much more sophisticated schemes can also be attained. The authors suggest a methodology for identifying which member of this family applies to the data under study, and then to calibrate the specific coefficients relevant to this model.

One useful substantive spin off from this method is that the number of weeks over which autocorrelation effects are still apparent will be found in the model identification stage. This could have an interpretation in terms of offender behaviour when viewed as a phenomena constrained by time and space.

Stage 1 involves find suitable p, d, q values. A common way of doing this is by estimating autocorrelations at various lags, and seeing how the values change as lag increases. The autocorrelation at lag k is defined as

$$\frac{\text{Covariance}\,(Z_t, Z_{t-k})}{\text{Variance}\,(Z_t)}$$

and the sample estimate of this is

$$\hat{\rho}(k) = \frac{\sum_{t=1}^{n-k} (Z_t - \bar{Z})(Z_{t+k} - Z_t)}{\sum_{t=1}^{N} (Z_t - \bar{Z})}$$

for each k. It can be shown that in the case where p=0 (purely moving average processes) then $\hat{\rho}(k) = 0$ for all k>q. This will clearly not hold exactly for the sample estimate, but a significance test may be performed, since asymptotically $\hat{\rho}(k) \sim N(0,1/n)$. Similarly, a test may be performed on sample partial autocorrelation estimates (based on the residual at lag k when other values have been allowed for). This decays exponentially (or is a sine wave whose amplitude decays) beyond lag q in the Moving average case. The converse holds for purely autoregressive models - the partial autocorrelation becomes zero if k>p and the ordinary autocorrelation decays. Finally, if p and q are both nonzero, both coefficients will eventually decay. Given this knowledge, sample autocorrelations and partial autocorrelations should yield some clues

as to p and q. It can also be shown that if d>0, autocorrelation will not tend to zero as k increases, as a deterministic trend will always contribute to correlation.

Therefore, inspection of these curves leaves the analyst with reasonable guesses as to p d and q. Next, consider the second stage of the process, where values of the regression coefficients are to be estimated. Several methods exist to do this, one of which is the conditional least squares method (BMDP manual 1985). In this technique the regression coefficients and the mean level are chosen to minimise

$$\sum_{i=1}^{n} \hat{\varepsilon}_i^2$$

where $\hat{\varepsilon}_i = b_1 \hat{\varepsilon}_{t-1} + Z_t - (1-q_1)Z_{t-1} + q_1(Z_{t-2})$

At this point, significance tests to see if any ais or bis should not have been included may be performed. Also, as estimated residuals now exist, it may be worth checking these for autocorrelation. If the model is good, there should be little evidence of this, as residuals should be independent. This constitutes the third stage of analysis, where the model arrived at is checked. From this, ideas as to p, d and q may be modified, and the sequence of stages repeated until a workable compromise is met. This should not usually take many cycles, perhaps two or three. At this point, the performance of competing models should not differ enough to merit further examination.

A methodology such as this is well established, and deserves consideration clearly, but there are some shortcomings. Firstly, it is based on the assumption of a stable relationship between the variable at time t and the values at t-1, t-2 and so on. The regression coefficients are fixed with respect to time. Possible changes in circumstance may occur in the study area, however, and these may cause changes at some point in the coefficients. Such changes may not be modelled in the basic Box-Jenkins approach. Secondly, parametric assumptions are made in the model, with respect the the error terms. These are assumed to be Normal, but this may not be the case. A non-parametric test may be a useful back up to the Box-Jenkins approach. Such a test is now proposed. A nonparametric test of autocorrelation can be based on a test of the null hypothesis that x(t-1) and x(t) are indepedently distributed with the same distribution function F. Define the statistic s(t) as

1 if $x(t)<x(t-1)$ and $x(t)<x(t+1)$

1 if $x(t)>x(t-1)$ and $x(t)>x(t+1)$ for t=2 ... n-1

0 otherwise

The sum from t=2 to n-1 of S(t) can be thought of as a count of peaks and troughs in the data, viewed as a sequence in time. It may be shown, (appendix 4.2), that if n is sufficiently large, this sum (call it U say) has an approximately Normal distribution with mean 2/3(n-2) and variance $\frac{5n}{18} - \frac{29}{90}$. Thus a test of the null

hypothesis of temporal independence may be carried out if U is computed. A lower-tail value of U is evidence of either a trend or positive correlation, whilst a large U corresponds to excessive peaking and troughing probably brought about by negative correlation or oscillations.

A final test is also proposed to check against the first phenomena of changes in the Box-Jenkins coefficients over time. To do this, the fairly crude technique of splitting the three years worth of data into single yearly subsets and analysing each in turn will be carried out. the results should be fairly consistent if the model is stable over time. However, if this does not appear to be the case, a more adaptable forecasting technique than Box-Jenkins should be used.

4.3.2 Results

Firstly, consider Box - Jenkins analysis applied to the entire time span of the data set. Figure 4.12 illustrates the autocorrelation function, whilst figure 4.13 shows partial autocorrelation. Notice that although the autocorrelation function is not significantly greater than zero after the first lag, the function appears to die out exponentially, at least until random noise dominates the high order lag estimates. Similarly, the partial autocorrelation, also significant only at lag one, appears to vary randomly beyond lag two or three, say. Given that a parsimonious solution is preferable, and

Figure 4.12
Correlogram for Household Burglary

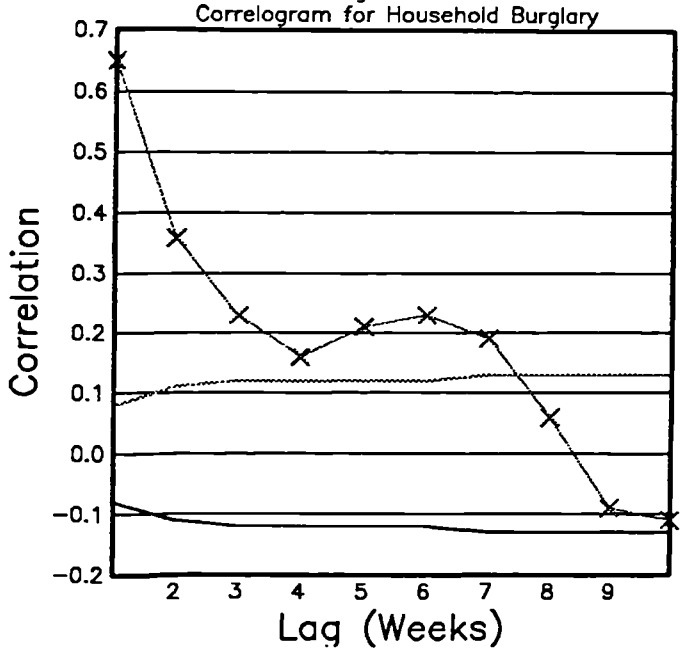
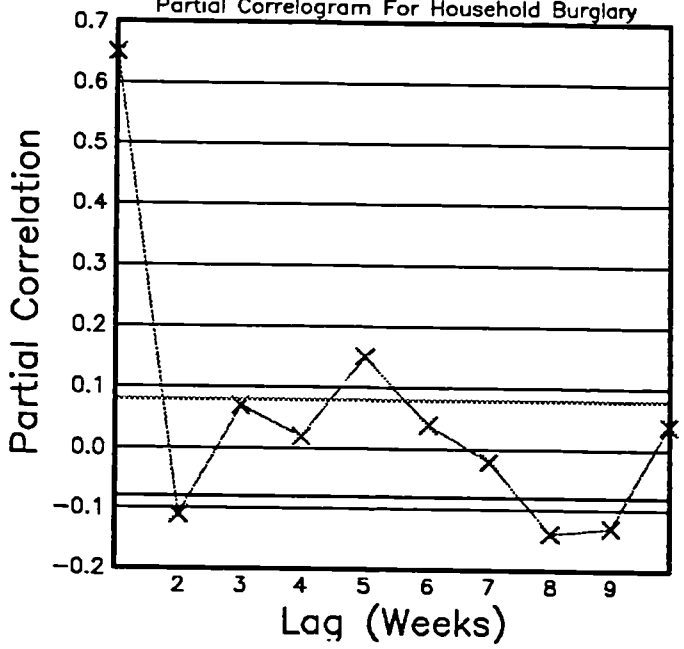


Figure 4.13
Partial Correlogram For Household Burglary



that any effects beyond where the correlogram estimates are swamped in white noise cannot be detected, it seems a reasonable first guess to model the process as as AR(1). This will be the point of departure in the Box-Jenkins modelling procedure.

At the next stage, three models will be fitted to the data. Firstly, an ARIMA(0,0,1), then (1,0,1) and finally (0,0,2). These can be seen as testing the model suggested above, and then testing the effect of adding either a second autoregressive term or a new moving average term to this model. The results are shown in table 4.7. Clearly, the moving average term is not necessary, and the second autoregressive term has only a minor effect. Therefore adoption of the initial model thrown up from correlation analysis seems reasonable. A final check may be to examine the residuals of the fit of this model for autocorrelation. If there seems to be none, it would suggest that these residuals are independent, as there are supposed to be in the model.

After this, the nonparametric test is carried out. Using the formula set out in appendix 2, as n=156, we have that (2/3)(n-2) = 105.67 as the expected value of peaks and troughs under a null hypothesis of independence. We also have that the the variance is 43.01 using the formula if n=156. This leads to a standard error of 6.6. The confidence limits are then 105.67+/- 1.96x6.6, roughly (92.7,118.6). The observed count is 87. This falls below the lower confidence limit of a two-tailed 5% significance test, and suggests the tendency for the data to peak or trough is less

Table 4.7

Box Jenkins Model Estimation: Conditional Least Squares Method

Model	Estimates	Variances	T-ratio
ARIMA(1,0,0)	AR(1)=0.9806	0.0163	60.25
ARIMA(1,0,1)	AR(1)=0.9864	0.0141	60.25
	MA(1)=0.1542	0.0816	1.89

than one would expect under an assumption of independence. This suggests either positive autocorrelation in the data, or a positive trend. Thus, the non-parametric results seem to add support to the conclusions based on Box-Jenkins analysis in the last paragraph.

Lastly, consider Box-Jenkins analysis on the data split into three separate years. Here the ARIMA(1,0,0) model is fitted to each year in turn. Estimates for the autoregression coefficient and the mean level are given in table 4.8. Simple significance tests may be carried out, between pairs of coefficient estimates, given asymptotic Normality of the estimation process. These must be considered fairly informally for two reasons.

1) Tests on different coefficients and for different time periods are not independent.

2) The estimates themselves are part of a time series related to the one used to model the data set.

They do, however, give some indication of consistency, or lack of it, between the coefficients when viewed as time progresses. It seems reasonable to compare the first and last year, as estimates based on these two years are least likely to exhibit correlation. If a1 for 1984 is estimated as 0.9955 with S.E. 0.0407 and a1 for 1986 estimated as 0.9963 then the difference between the estimates for the

Table 4.8

Consistency Of Box-Jenkins Model Over Time

Year	AR(1) Coefficient	Variance
1	0.9955	0.0407
2	0.9851	0.0229
3	0.9963	0.0149

two years will be distributed approximately Normally with mean 0 and S.E.

$$\sqrt{SE_1^2 + SE_2^2}$$ where SE_1 and SE_2 are std. Errs of each estimate.

under the hypothesis that there is no difference between them, and assuming approximate independence of the two estimates. Since a1986-a1984 = 0.0008 and the combination of the standard errors using the formula above gives 0.0443, the standardised Normal test variate is 0.1806. This is not significantly different from zero at the 5% level of significance (for a two-tailed test the lowest significant value is 1.96). Thus, evidence here suggests that a simple autoregressive prediction scheme should work adequately with the subdivisionally aggregated data.

4.4 Space-Time Models

Until this stage, the model for crime rate variation has been considered either as a process in space aggregated over time, or as one evolving in time but totalled over space. While analysis of the data from each of these two perspectives is informative, greater realism may be achieved if the process is considered to be simultaneously referenced in both space and time. If it can be stated that relatively little work has been applied to the problem of spatial stochastic processes, then the scarcely of work in space-time processes is an order of magnitude greater. Various techniques do exist in terms of multivariate time-series analysis, treating each beat as a time-series interacting with all of the other time series in the system.

Such a view of the system is reasonable, but there is a danger of ignoring the geographical aspects of the process. The crime rates of certain beats may correlate highly to those of others for spatial reasons (ie there will be a tendency for adjacent beats to have correlated rates), and modelling should allow for this. It therefore seems appropriate to attempt some specific analysis of space-time dependency before fitting multivariate time series models.

In the case of household burglaries, there is strong informal evidence that incidents occur in 'epidemics' or 'clusters'. In more detail, the likelihood of a burglary occurring at a given address at a given time is thought to increase if addresses in the locality have recently experienced burglary. Although based on different causal assumptions, similar quantitative models may be applied to infectious disease epidemics, and several statistical tests for 'epidemicity' exist. The purpose of these tests is to determine whether space-time dependency of the kind described above exists, and if it does, to what scale of time and distance. Therefore, as an initial stage of space-time analysis, tests of this sort will be applied to the household burglary data. The results of these tests may then be used as input to the building of multivariate time series models whose correlation structure properly reflects the local geography of crime incidence.

After this stage, the consideration of several possible space-time series models will take place, allowing a decision to be made as which is the best predictor. Each model will be calibrated on a

training data set, and then applied to further data, allowing the quality of its performance to be assessed. The conclusion of this study should yield a model suitable for incorporation into the crime pattern analysis system.

4.4.1 Analysis Of Space-Time Interdependence : The 'Epidemic' Effect.

As mentioned above, tests have been developed for space-time interaction, so that processes in which an 'Epidemic' effect occurs may be identified. The most conceptually simple test of such clustering is that initially proposed by Knox (1964) which the number of event pairs that are close in both space and time are counted, and the significance of this count is tested against the distribution of such a count under the null hypothesis that the distributions for spatial and temporal referencing of events are independent. Closeness in space and time are defined by the experimenter, usually in terms of Euclidean distance and absolute time difference. When the both the distance and elapsed time between an event pair do not exceed certain values set by the experimenter, (termed the critical time and critical distance) the pair is said to be 'close', and the test statistic is defined as the count of all close pairs in the data set.

Knox suggested using a Poisson approximation for the distribution of this statistic, which works well if the proportions of space-close and time-close events are small. However in the case of the crime data, when burglaries have been reported for several adjacent postcode units and on most days of the year, relevant definitions of closeness may well give larger proportions than would be suitable for this sort of approximation. Thus, an additional strategy will be adopted. Under the hypothesis that spatial and temporal distributions are not related, any permutation of grid reference-day of year pairings are equally likely. Thus, if each possible Knox count for each permutation is evaluated and these are sorted, the observed count may be compared against this list. Since each value is equally likely, the ranked list of possible values of counts gives a list of n-tiles of the null randomisation distribution, so significance testing may be carried out. However this would require n! evaluations of the Knox statistic, and the computation time required would be impractical. Thus 99 permutations will be generated randomly and the test statistic compared against this. It may be shown that the exact significance of the statistic is its rank when added to the 99 simulated results (Hope, 1968). This practice is known as Monte-Carlo testing.

Knox test provides a test of the hypothesis stated below:

$$h0 : f(space,time) = f1(space)f2(time)$$

where $f, f1, f2$ are probability density functions and t is the time of an event, x is the position of the event, expressed as a two dimensional vector. Tests of this hypothesis may be generalised by introducing the test statistic

$$\sum_{i,j} X_{ij} Y_{ij}$$

where $X = f(d_{ij})$

and $Y = g(t_{ij})$

Here, Knox's test is obtained by putting

$f(x) = 1.0$ if $d < c_d$

$= 0$ otherwise

$g(x) = 1.0$ if $t < c_t$

$= 0$ otherwise

However, using these particular f and g values excludes any evidence of clustering at scales exceeding the critical time and distance. This could be thought of as reducing the power of the test - a bad specification on the part of the experimenter may stop only a slightly weaker clustering process from being detected, since weighting of any close events exceeding the critical time and distance is zero. Mantel (1967) proposes tests in which f and g are monotone decreasing decay functions, rather than abrupt step cutoff functions.

An earlier idea of Mantels was to use $\Sigma t_{ij} d_{ij}$ itself. In this case, testing for clusters would take place in the lower tail of the distribution of the test statistic. However, this was thought to be problematic, as the greatest weighting would be given to events least close in space and time, which are unlikely to exhibit correlation even if some space-time clustering does occur. (It might, however, be a good statistic for 'repulsive' clustering, when an event occurring at a certain ($\underline{x}$,t) point inhibits similar events near to it for a time period, although even this may be inhibited if the data covers a very large expanse of space and time, on a much larger scale than the scale of space-time repulsion.) Thus, the result in this case would be a reduction in sensitivity to cluster detection, or a loss of power in the testing procedure. This leads to the decision to use monotone decreasing f- and g- functions. Previous simulations have shown that functions of the form

$$f(d_{ij}) = (d_{ij} + \alpha)^{-1}$$
$$g(t_{ij}) = (t_{ij} + \beta)^{-1}$$

are most sensitive (Siemiatycki, 1978). It is therefore proposed to run further Monte Carlo tests using the above statistic, in addition to the Knox tests. Here, a = 100 and b = 1. In order to reduce computing time, beyond certain day-gaps (over 7 days) the weighting will be uniformly zero, although it decays smoothly up to this point. However, all levels of distance will be non-zero weighted. Since the type of clusters being sought are less than weekly gaps, this is unlikely to cause problems.

4.4.2 Accuracy Of Spatial Referencing

Another aspect of this kind of test is connected to the precision of the spatial referencing. For the post-coded household burglary data, 8-digit grid referencing is used. This specifies easting and northing to the nearest 100m. In the vector notation, each element of $\underline{x}$
is coded to 4 significant digits. It is possible that Knox testing will be carried out for critical distances of this order of magnitude (say between 100-500m). Since there will be an associated uncertainty of +/- 50m to each reference (assuming grid references are rounded), the effect of 'wobbling' the grid centroids should be examined, to discover to what extent the Knox statistic and its Monte Carlo significance will be affected.

It may also be important, particularly in rural areas, to consider the effect of spatially referencing events by postcodes. In

less densely populated regions inter-household distances may be large, and a single postcode could cover quite a large region. This may tend to pull spatially dispersed phenomena together, or at least give this appearance in the database. This effect may also be analysed.

Turning attention initially to the former of the two problems, simulation again seems to be the only reasonable course to take. Several estimates of Knox's statistics may be computed from several 'wobbled' data sets. The procedure consists of adding a uniformly distributed distance in the range [-50,50] metres to each easting and northing in the data set, and then computing Knoxs' statistic. This process is repeated several times (say 100). Thus each 'wobbled' dataset could have produced the final, rounded data set that is actually recorded in the data base. Each simulated data set may be thought of as having been drawn from an infinite pool of possible exact data sets, and each of these datasets as having an associated Knox statistic. Thus, from modelling the uncertainty in the true dataset, the uncertainty in the Knox statistic can be investigated. A sample of Knox statistic values can be generated and from this approximate confidence limits can be computed. This process need only be done for critical distances near to the 100m level, as for higher distances the relative effect of rounding will be small.

Ideally, in addition to gaining an approximate distribution for the Knox statistic it might be useful to perform a randomisation test on each of the 'wobbled' data sets, to see if there is any uncertainty in the significance results obtained on the rounded data. However it is

feared that if this is done on all 100 simulated sets, the cost in computing time will be too great, requiring 100x100 computations of Knox statistics. More informally, however, it will be possible to examine the test results for a handful of 'wobbled' sets.

The problems associated with errors due to spatial referencing by nearest postcode centroid will now be considered. In urban regions, nearer to the city centre, it may be noted that the distance between postcode centroids is near to 100m, so that every possible 100m rounded point is used. In these cases, the effects of wobbling may be regarded as similar to postcode rounding. Problems arise, however, in remote areas. For example, in beat W3 a few addresses are of isolated houses sharing postcodes with their nearest neighbours, but the distance between these neighbours greatly exceeds 100m. A possible means of addressing this problem is to run further simulations allowing greater variations for certain postcodes. A simple way of achieving this is to allow different levels of uncertainty for each postcode sector, based on mean distances between households within the sector. These mean distances will be based on square roots of mean areas occupied by individual houses. Since approximations of the number of houses and the areas of postcode sectors exist, these figures may be computed.

4.4.3 Results

The results of the Knox tests are listed in table 4.9. These are based on the 100m grid references from postcode centroids. Four main tests are carried out. The first of these is for a critical time of 1 day, at a critical distance of 200m. This is designed to be sensitive to short-term time clusters, separated by about two postcode units. Thus, almost daily epidemics at a 'within neighbouring streets' level of separation is being investigated. Then the distance is increased to 3km, which is roughly the average beat separation distance based on household centroids. These two critical distances are then applied in turn with a critical time of one week, this being the anticipated horizon for forecasting in a working system. Clearly, the observed figures are all highly significant, showing a much larger tendency for events close in time and space to occur than could be attributed to chance. In addition to this, Mantel-type statistics are computed as specified in 4.4.2. The results are listed in table 4.10. Again, results appear highly significant.

Having carried out these tests, the 'wobble' test results must now be considered. The initial simulation, investigating the *variation of the knox statistic under rounding*, is summarised in table 4.11. Note that the count of space-time close events for wobbled data has a marked tendency to exceed that of the rounded data for critical distance 200m. This appears to occur also with Mantels statistic.

Table 4.9

Results Of Knox Tests

	---- Poisson Model ---			---- Randomisation ----			Closeness	
k	E(k)	SD(k)	Sig.	E(k)	SD(k)	Sig.	Days	Dist.
460	222	7.73	0.000	222	7.72	0.000	1	200m
1505	1092	14.91	0.000	1092	15.39	0.000	7	200m
10407	10211	33.04	0.000	10204	37.30	0.000	1	3km
51294	50516	101.05	0.000	50163	80.40	0.000	7	3km

N.B. k is the number of events close in both space and time as set out in the "closeness" columns.

Table 4.10

Results Of Mantel Test

	Under randomisation:		
Observed	Mean Value	St. Dev.	Signif.
1625.41	1378.71	11.08	0.000

Table 4.11

Sensitivity Of Knox Tests To Variability Of Spatial Referencing

Closeness			Distribution of k under "wobbling"			
Days	Distance	True k	Mean	S.D.	Min	Max
1	200m	460	532.6	8.65	510	554
7	200m	1505	1840.0	25.58	1779	1907
1	3km	10407	10450.1	10.70	10422	10479
7	3km	50516	51468.0	33.07	51359	51559

N.B. k defined as in table 4.10

Table 4.12

Sensitivity Of Mantel Tests To Variability Of Spatial Referencing

Mantels Statistic : Variation under "wobbled" data

True Val	Mean	S.D.	Minimum	Maximum
1625.41	1547.44	2.62	1540.92	1553.45

It seems that results provide sufficiently extreme evidence in favour of clustering that the extra uncertainty in wobbled data cannot detract from this. In all of the wobbled cases, the reduction in Knox's statistic due to rounding applies equally to true and randomised data sets, so that ranking is virtually unaffected, and significance levels remain stable.

Finally, the above techniques are applied to Mantels test. Due to the increased computational overheads in computing mantels test, only 9 simulations are carried out. However, as may be seen in table 4.12, similar conclusions may be drawn.

4.4.4 Conclusions

Firstly, there is strong statistical evidence from the tests applied to this data that space-time 'epidemics' in this data do occur. This effect is apparent at both a day-to-day neighbourhood level, and on a week-by-week inter beat basis. Both of these conclusions may be put to good use in a crime pattern analysis system. The former could be used to interpret past data; those cases which are within critical time and distance of each other could be highlighted on a VDU graphics map, indicating potential clusters. If the crime records for the highlighted points are consulted, subjective analysis of incidence descriptions and modus operandi might lead to patterns being identified.

On a predictive basis, beatwise crime rates tabulated by week may be used in a space-time autoregression (STAR) model or similar. The larger scale spatial scale of aggregation is required here to reduce computational tasks when making predictions: analysis accurate to postal code units means forecasting requires the analysis of some 1200 spatially autocorrelated time series, which would be a daunting task for currently available hardware, to say the least! However, the previous analysis indicates that there is sufficient space-time interaction between beats on a week to week level of separation to suggest that beatwise time series STAR predictors (or similar) will produce fruitful results.

Consider now the results of the 'wobbling' simulations. As discussed, the result of 'wobbling' has the effect of increasing Knox scores. A possible explanation for this is that rounding has the effect of forcing the spatial referencing onto a lattice whose points are allocated at 100m easting and northing intervals. Thus, a 'repulsion' effect is introduced detracting from the spatial autocorrelation effect. This effect becomes negligible for larger distances, as the interval of the lattice becomes relatively small, making it virtually 'cover' the region under examination. However, for critical distances close to 100m, the effect is notable.

The above argument applies mainly to rounding errors, but errors due to assignment to nearest postcodes must also be thought of. The simulations applied to this problem yield similar results to those purely based on rounding. This is most likely explained by the fact

that the majority of postcodes are within 100m of each other, providing a similar partitioning of the study region to the 'rounding zones'. The relatively small number of rural postcodes do not strongly affect the overall measure of clustering. However, since assignment inaccuracy of this type is more likely to distort rural data, a cautious conclusion to be drawn from the 'epidemic' tests might be that there is evidence for this type of clustering in urban regions (or more formally, in densely populated areas).

When considering both of the above effects with specific regard to Knox's statistic, it is relatively easy to understand the effects. As the contribution to Knox score is constant (for fixed time intervals) for all distances below the critical, and uniformly zero above this, a pair of close events could be separated by rounding error (if each one lies on opposing sides of a rounding zone border) reducing their effect on the overall statistic to zero. If they are brought closer together (ie onto the same rounding lattice point) this would not alter their Knox score contribution as they are already close. Thus, the overall effect of rounding is to reduce the statistic. A converse effect could also occur for rural events which may not be close, but are brought together by mapping onto the same postcode unit centroid. However, for this data, the incidence of this is low, so this effect will be superseded by the former. Thus it is reasonable to conclude for this data set that greater accuracy in the spatial referencing of events may lead to even more significant results than those obtained here.

4.4.5 A Space-Time Prediction Model

In the last section, tests have led to the conclusion that the spatial distribution of household burglaries in the study area interacts strongly with their distribution in time. Events of up to a week in the past have some effect on the likelihood of events occurring in the present. Clearly, then, there is a reasonable basis for the short-term forecasting of the geographical distribution of household burglary rates into the future, using this data. The aim of this section is to devise a model, linking the geographical distribution of past data to that of the following week; that is, to model the data as a space-time series. As discussed previously, to limit computational overheads it will be best to analyse data aggregated to foot beats. However, it is hoped that any space-time models used will be general to any areal unit, so that at some point in the future, should advances in computer hardware permit, the method may be implemented to a scale of greater resolution, for example to post code unit areas. Thus, it is required to develop a stochastic model of crime rates in each beat on a week-by-week basis. This may be achieved by borrowing from both the spatially autocorrelated model of Moran and from Box-Jenkins style time series modelling.

It has already been noted that relatively little work has been done in the area of space-time modelling, and some of the model fitting techniques may appear to be 'ad hoc' in nature. One of the principal aims of the PhD. is to create a prototype predicting system for

evaluation, so that it seems important to have some form of fitted model, perhaps only as a first approximation, rather than to get side tracked into a parallel problem. However, the design philosophy of the final system is modular, so that any model calibration improvements to be discovered in other research could eventually replace my initial efforts.

4.4.6 Model Specification

Several prototypes for space-time models exists, and in particular a family of space-time models analogous to the Box-Jenkins framework for time series modelling may be put forward. To simplify matters, the moving average component could be dropped to give models of the form

$$\underline{X}_t - \underline{\mu} = \sum_{i=0}^{n} C_m (\underline{X}_{t-n} - \underline{\mu}) + \underline{\varepsilon}$$

$$\underline{\varepsilon} \sim N(0, \sigma)$$

In these models, space-time stationarity must be assumed. This is defined by Bennett, 1979.

This makes it necessary to standardise each element of $\underline{X}$ by subtracting its mean value. The (k,j)th element of the matrix reflects the influence that $X_{k,t}$ has on $X_{j,t+1}$. In a process such as

this, in which spatial autocorrelation is expected to occur, the specification of the elements of A should reflect the spatial structure of the areal units. It seems reasonable to fit a distance decay function to the elements of A, thus:

$$C_{ij} = a f(d_{ij})$$

where dij is a distance measure between beats j and k (Choices of distance measure have been discussed in 4.2), and f is a monotone decreasing function, such as an inverse power or negative exponential function. Considerable computing time will also be saved if the effects of second-order adjacencies, and orders beyond these are ignored. This is reasonable, when considering the investigation for the purely spatial model earlier in this chapter.

The model can be further simplified if the term in Xt is dropped from the right hand side of the equation. This effectively makes the elements in Xt conditionally independent, given the value of Xt-1. This will simplify the model calibration, and in a final predictive application will obviate the need to solve simultaneous linear equations in order to determine the forecasted values. An assessment of the loss in accuracy due to this simplification will be carried out, and if this loss is not great, predictions will take this form, to further the cause of obtaining a parsimonious predictive model.

4.3.7 Coefficient Estimation

Attention will now be turned to calibrating models of the kind discussed above. Firstly consider the calibration problem in which each term is independently distributed. Suppose that the usual square root transformation has been applied to the burglaries per household figures, and that further to this, they have been standardised about their respective means. Then, the conditional likelihood of an observed vector X at time t is

$$\exp\left[\left(\underline{X}-\underline{\mu}-C\left(\underline{X}_{t-1}-\underline{\mu}\right)\right)^{T}\Lambda\left(\underline{X}-\underline{\mu}-C\left(\underline{X}_{t-1}-\underline{\mu}\right)\right)\right]$$

Then the likelihood of an entire crosstabulation table of beats by weeks (conditional on the values observed at week one) is given by

$$L\left(\underline{X}_t \mid t=2..n\right) = \prod_{t=2} L\left(\underline{X}_t \mid \underline{X}_{t-1}\right)$$

$$= \prod_{t=2} K\sigma^{nm}\exp\left[\left(\underline{X}_t-C\underline{X}_{t-1}\right)^{T}\left(\underline{X}_t-C\underline{X}_{t-1}\right)\right]$$

and so

$$= K\sigma^{-nm}\exp\left[\left(\underline{X}_t-C\underline{X}_{t-1}\right)^{T}\left(\underline{X}_t-C\underline{X}_{t-1}\right)/2\sigma^2\right]$$

$$\therefore\ l\left(\underline{X}_t \mid C,\sigma\right) = \ln\left(L\left(\underline{X}_t\right)\right) = nm\ln\sigma + \Sigma\left(\underline{X}_t-C\underline{X}_t\right)^{T}\left(\underline{X}_t-C\underline{X}_t\right)/2\sigma^2$$

$$\therefore\ \left.\frac{\partial l}{\partial\sigma}\right|_{\hat{\sigma}} = 0 \Rightarrow \hat{\sigma}^2 = \frac{\Sigma\left(\underline{X}_t-\hat{C}\underline{X}_{t-1}\right)^{T}\left(\underline{X}_t-\hat{C}\underline{X}_{t-1}\right)}{n}$$

Now suppose A is parametrised by a parameter vector t. Then, the value of t maximising l, that is, the maximum likelihood estimate of t, satisfies

$$\left.\frac{\partial l}{\partial\underline{\theta}}\right|_{\hat{\underline{\theta}}} = 0 \Rightarrow \Sigma\left(\underline{X}_t - C\left(\hat{\theta}\right)\underline{X}_{t-1}\right)^{T}\left(\underline{X}_t - C\left(\hat{\theta}\right)\underline{X}_{t-1}\right) = 0$$

Thus t minimises the least squares error of Axt-1 when viewed as a predictor of xt. Note that once this value, from now on referred to as SS, has been computed, it can be shown that

$$\hat{\sigma}^2 = \frac{SS}{nm}$$

Suppose, for example that a model of the form

$$\underline{X}_t = C\underline{X}_{t-1} + \underline{\varepsilon}$$

$$\text{with} \quad C_{ij} = c\, d_{ij}^{-\alpha}$$

is proposed, then the MLEs of c and alpha would be given by

$$\underset{c,\alpha}{\operatorname{argmin}} \sum_t \left| \underline{X}_t - C\underline{X}_{t-1} \right|^2$$

Note that this problem cannot be solved explicitly, and that often least squares problems require iteratively. However, several numerical algorithm libraries exist enabling numerical solutions to be computed.

Finally, it may also be noted that likelihood ratio tests may be perform on the curve fits, to test null hypotheses referring to simplifications of the parametric form of A. For example, to test whether separate values of C_i should be given for each beat in a model of the form

$$C_{ij} = C_i\, d_{ij}^{-\alpha}$$

then

$$-2\ln\left(\frac{L(\underline{X}|C,\alpha)}{L(\underline{X}|C_i\forall i,\alpha)}\right) \sim \chi^2_K$$

where k is the difference in degrees of freedom of the two models.

The more complicated case occurs when the term in X is no longer assumed to be independent. In this case, the multiplicative model for the likelihood of a given realisation is no longer correct, as the deviations about the the means for each beat are now correlated. One possible solution approach to estimation in this case is to use 'pseudo-likelihoods'. In the purely spatial case, these have been used by Besag (1975) to estimate parameters in an autoregressive model. In the purely spatial case, pseudo-likelihood is defined as

$$\prod_{i=1}^{\#\text{beats}} P(X_i \mid X_{\delta/i}, \theta)$$

where δ/i denotes the set of neighbours of area i.

Thus, true likelihoods of each event are replaced by the products of the conditional likelihoods of each observed beat rate given the rates of its neighbours, and spatial autocorrelation in these conditional

distributions. In this space-time model it is proposed to extend this idea into the time dimension thus :-

$$\prod_{i=1}^{\#beats} \prod_{j=1}^{\#weeks} P(X_{ij} \mid X_{\delta(i,j)}, X_{\delta(i,j-1)})$$

with, for example,

$$X_{ij} \sim \text{Normal}(X_{ij} \mid X_{\delta(i,j)}, X_{\delta(i,j-1)}, \underline{\theta})$$

Then, given the Normal distribution model, the pseudo-likelihood of the entire beat by week crosstabulation will be given by

$$\prod_{i=1}^{\#beats} \prod_{j=2}^{\#weeks} \exp\left(\tfrac{1}{2}\left(X_{ij} - \sum_{k \in \delta(i)} a_k X_{kj} - \sum_{k \in \delta(i)} b_k X_{k,j-1}\right)^2 / \sigma^2\right)$$

and again this is minimised by a least squares fit, but regressing the values of a beats' neighbours onto itself in addition to the values lagged by one or more weeks. When forecasting is being carried out, this leaves a simultaneous equation in the predicted values to be solved:-

$$E(\underline{X}_{j+1}) = AE(\underline{X}_{j+1}) + B\underline{X}_j$$

whereas exclusion of unlagged autocorrelation yields predictions directly as a linear transformation of the observed lagged crime rates.

4.4.8 Choosing Models To Be Fitted

Until now, the fitting of STAR models has been considered only for the general case of parametrising the regression matrices Ai. In this section, a set of specific parametrisations will be proposed. All of these are intended to reflect the spatial and temporal structure which examination of the data up to now has suggested occurs. Some of the models specified will be relatively simplistic, just having a general power law relating the inter-beat distance with the regression coefficients, while other will be more complex, allowing for different beats having different sensitivity to phenomena in surrounding beats, also allowing for non-lagged autocorrelation.

A)

$$E(X_{i,j+1}) = \sum_{k \in S/i} \left(d_{ik}^{-\alpha} C X_{kj} \right) + \alpha X_{ij}$$

Here, no spatial autocorrelation at lag zero is assumed, and all beats are assumed equally sensitive to neighbouring crime rates.

B)

$$E(X_{i,j+1}) = \sum_{K \in \delta/i} (d_{iK}^{-\alpha} C_i X_{Kj}) + d X_{ij}$$

In this case, sensitivity to adjacent beats varies throughout the subdivision, but no allowance for lag zero correlation is made.

C)

$$E(X_{i,j+1}) = \sum_{K \in \delta/i} (d_{iK}^{-\alpha} (C_1 X_{Kj} + C_2 X_{Kj+1}) + d_1 X_{ij} + d_2 X_{ij-1}$$

As A) but considering events at time lags of two weeks also.

D) $$E(X_{i,j+1}) = \sum_{K \in \delta/i} (d_{iK}^{-\alpha} (C_0 X_{Kj+1} + C_1 X_{Kj})) + d X_{ij}$$

As A) but allowing for zero lag correlation.

E)

$$E(X_{i,j+1}) = \sum_{K \in \delta/i} (d_{iK}^{-\alpha} (C_{0i} X_{Kj+1} + C_{1i} X_{Kj})) + d_i X_{ij}$$

As D), but allowing sensitivities to vary from beat to beat.

4.4.9 Results

For each of the five models suggested in the last section, coefficients are estimated using the least squares technique. The crime count variables are subject to the usual correction for household density and square root transforms. The goodness of fit results are listed in table 4.13. These are fitted to data with the mean levels extracted, so that the number of parameters refers only to those in the autoregression formulae. A notable change in least squares fit occurs when moving from the base model (ie when only mean values for each beat are fitted) to the simplest correlated model (model A). A second notable jump occurs when terms for zero lag spatial autocorrelation are incorporated (models D and E).

In addition to simply measuring the descriptive index of least squares fit, some likelihood ratio tests may be carried out. These allow significance testing of some relevant hypotheses. Using the likelihood ratio formula set out in the last section, it can be shown that for

Table 4.13

Goodness-Of-Fit Of Space-Time Autoregression Models

Model Code (see Text)	Sum Of Squares	No. Parameters
A	1.2432	2
B	1.2275	31
C	1.2320	3
D	1.1793	4
E	1.1778	61

Table 4.14

Likelihood Ratio Tests

Hypothesis	Likelihood Ratio	D.F.	99% Point
Base vs A	420.25	2	9.21
A vs B	59.47	29	49.6
A vs C	42.35	1	6.63

models A-C, and for the base model, ratio test statistics for pairs of models as competing hypotheses take the form

Model 1 (n_1 parameters) Vs Model 2 (n_2 parameters)

which are asymptotically distributed as $-2Ln\left(\frac{L_2}{L_1}\right) \sim \chi^2_{n_1 n_2}$ where n_1 and n_2 are the number of parameters in each model. Clearly, and without loss of generality, model 1 must be taken to be that with the greater number of parameters. Models D and E cannot be subject to testing of this sort, unfortunately, since they are not calibrated in maximum likelihood terms.

Test results are listed in table 4.14. There is clearly strong evidence for some autocorrelated model against the base model, since the test statistic is about 200 times the mean of its null distribution!
In addition to this, both models B and C outperform A. The drop in sums of squared error in model D suggests that this is notably different from these models also. However, the marginal improvement of E on D suggests that E does not notably outperform D.

These results suggest once again that there is a significant space-time effect in the occurrence of household burglaries. It appears that the best models are those capable of incorporating the effects of both lagged spatial autocorrelation effects and synchronous effects at the time over which burglary rates are to be predicted. It must be borne in mind, however, that these results are eventually destined to be

part of a desktop micro forecasting system, and therefore the complexity of the synchronous correlation may prove difficult and time consuming to evaluate in a working system (this is discussed in the light of the final Bayesian prediction model in the following chapter). However, the best performing of the remaining models offers a significant improvement on the base level model, and should provide a relatively easily programmable solution to the problem of crime prediction which is reasonably effective.

4.5 Conclusions

A recurrent result in all of the fitting of spatial probabilistic models in this chapter is that there is a strong space and time interaction effect in the data, and that due to the autocorrelation effect of this, records of space and time referenced household burglaries provide useful information when predicting future crime rates over several regions. In fact, latter results suggest that these alone can be the basis of a prediction system. This is strong evidence for the feasibility of an automated crime forecasting and analysis system that may be implemented at subdivisional level. As suggested in chapter 2, data consisting of spatially and temporally referenced crimes would be readily available in the everyday workings of a police station, so that the "housekeeping" of the data set on a day by day basis could be easily implemented.

Another important point is that some of the analysis techniques covered in the earlier parts of the chapter may also be applied to this data; namely the kernel estimation and Knox testing procedures. As recorded earlier, one police officer, on seeing the surfaces obtained from kernel estimation found the representation more readily interpretable than say, scatter plots or bar histograms of the data. Another found the Knox testing idea particularly relevant, pointing out that their personal method of crime pattern analysis was to look out for clusters of events that were close in space and time. Thus, these methods yield helpful methods of past data presentation which may also be of aid to crime pattern analysis. Since the data required for these techniques will already be fed into the system for predictive purposes, it would clearly be beneficial to incorporate the techniques as options in the prototype system.

LISTINGS FOR CHAPTER 4

```
C
C            ********* Listing 4.1 **************
C
C
C    Kernel Estimation Program in two dimensions ----
C      Written for police crime incidence data ------
C        Automatically attaches files
C
C             Channels 1 = CRIMESPOTS (point locations of crimes)
C                      2 = KERNMAT    (30x30 Kernel estimation of PDF)
C
C
      REAL*4 KERNEL(30,30), X, Y, K, BANDWT
      INTEGER I, J
      DO 50 I = 1, 30
        DO 50 J = 1, 30
   50     KERNEL(I,J) = 0.0
C
C  Define the Kernel Size
C
      WRITE (6,'(19H&Enter Bandwidth > )')
      READ  (5,*) BANDWT
      K = BANDWT * BANDWT
      BANDWT = BANDWT / 4.0
C
C  Attach Relevent Files
C
      CALL SETLIO(1,'CRIMESPOTS ')
      CALL SETLIO(2,'KERNMAT ')
C
C  Begin the main loop
C
  110 READ (1,'(T3,2F4.0)',END=120) X, Y
C
C  Convert to array parameters
C
      X = (X - 4180.0)/4.0
      Y = (Y - 5650.0)/4.0
      IMIN = INT(X - BANDWT)
      IMAX = INT(X + BANDWT + 1)
      JMIN = INT(Y - BANDWT)
      JMAX = INT(Y + BANDWT + 1)
      IF (IMIN .LT. 1) IMIN = 1
      IF (IMAX .GT.30) IMAX = 30
      IF (JMIN .LT. 1) JMIN = 1
      IF (JMAX .GT.30) JMAX = 30
C
C  Fit the kernel
C
      DO 100 I = IMIN, IMAX
        DO 100 J = JMIN, JMAX
          HUMP = 1 - ((X-FLOAT(I)-0.5)**2 + (Y-FLOAT(J)-0.5)**2)/K
          IF (HUMP .GT. 0.0) KERNEL(I,J) = KERNEL(I,J) + HUMP
  100     CONTINUE
      GO TO 110
  120 CONTINUE
      SUM = 0.0
      DO 130 I = 1, 30
        DO 130 J = 1, 30
  130     SUM = SUM + KERNEL(I,J)
C
```

```
C  Make it a distribution
C
      DO 140 I = 1, 30
        DO 140 J = 1, 30
  140     KERNEL(I,J) = KERNEL(I,J) / SUM
C
C  Output Coordinates
C
      DO 150 J = 30, 1, -1
        DO 150 I = 1, 30
  150     WRITE (2,*)  I*4+4182.0, J*4+5652.0, KERNEL(I,J)
      STOP
      END
```

Appendix 4.1

Calculation of Approximate Confidence Regions

Consider an approximate 5% confidence region R such that

$$f(x) > f(y) \quad \forall\, x \in R,\ y \notin R$$

Although an exact functional form for f does exist, namely the kernel estimate as an algebraic expression (see main part of chapter), a faster method would be to calculate values of this function on an n x m grid of arguements and multiply each by the grid square area. This will only approximate the integral over a set of square, but should be adequate if F is reasonably smooth. When such an array of values has been computed, it may then be sorted in descending order. Denote the kth element in the one-dimensional representation of this array as x_k and let i(n) and j(n) be the original coordinates of the element in the m x n grid before sorting. Then find K such that

$$\sum_{i=1}^{K} x_i > 0.05 \ ; \quad \sum_{i=1}^{K-1} x_i \leq 0.05$$

The set of grid squares such that

$$x_{i(l),j(l)} \in \{x_1 \cdots x_K\}$$

are then an approximation of the required area . Alternatively

$$\{x_1 \ldots x_{K-1}\}$$

ma y also be considered as an approximation, as a lower limit. From these, either F could be estimated as the value x_K divided by the grid square area, and this could be given to a contour drawing program, or the squares themselves could be coded and fed to a raster based program to map the relevant zone.

Appendix 4.2

The Peak Count Test Statistic

Consider a data set {X} with n observations, with each X independently and identically distributed (IID). For k = 2 to n-1, let i_k be defined as

$$1 \text{ if } (x_k > x_{k-1} \text{ and } x_k > x_{k+1}) \text{ or } (x_k < x_{k-1} \text{ and } x_k < x_{k+1})$$

$$0 \text{ otherwise}$$

Then, the peak count statistic is

$$\sum_k i_k = P$$

and thus

$$E(P) = \sum_k E(i_k) = (n-2)\,E(i_k)$$

but

$$E(i_k) = Pr(i_k = 1) = \frac{2}{3}$$

$$\therefore E(P) = \frac{2}{3}(n-2)$$

If x_{k-1}, x_k, x_{k+1} and are IID, then any ordering of these is equally likely. Replacing by ranks, there are 4 arrangements out of a possible 6for which $i_k = \underline{1}$: See below

x_{k-1}	1	1	2	2	3	3
x_k	2	3	1	3	2	1
x_{k+1}	3	2	3	1	1	2
i_k	0	1	1	1	0	1

Also,

$$\mathrm{Var}(P) = \sum_{K} \mathrm{Var}(i_K) + \sum_{\substack{K,j \\ j \neq K}} \mathrm{Cov}(i_K, i_j) \qquad (1)$$

If $|j-K|>2$ then $\mathrm{Cov}(i_j, i_K)=0$ (2), because no element of $\{x_{j-1}, x_j, x_{j+1}\}$ is correlated with any of $\{x_{K-1}, x_K, x_{K+1}\}$.

Now $$\mathrm{Var}(i_K) = E(i_K^2) - (E(i_K))^2$$

$$= E(i_K) - (E(i_K))^2$$

$$= \frac{2}{3} - \frac{4}{9} = \frac{2}{9}$$

and $$\mathrm{Cov}(i_{K+1}, i_K) = E(i_{K+1} i_K) - E(i_{K+1}) E(i_K)$$

$$E(i_{K+1}) = E(i_K) = \frac{2}{3}$$

$$E(i_{K+1} i_K) = \Pr(i_{K-1} = i_K = 1) \qquad (3)$$

The probability (3) is the probability that a pair of consecutive peaks in a sequence of four numbers appear. This is computed by direct enumeration:

In $4! = 24$ combinations, 10 satisfy the condition.

$$\therefore \quad Pr(i_{k-1} = i_k = 1) = \frac{5}{12}$$

$$\therefore \quad cov(i_k, i_{k+1}) = \frac{5}{12} - \left(\frac{2}{3}\right)^2 = -\frac{1}{36}$$

Finally $cov(i_{k+2}, i_k) = E(i_{k+2} i_k) - E(i_{k+2}) E(i_k)$

$$E(i_k) = \frac{2}{3}$$

$$E(i_{k+2} i_k) = Pr(i_{k+2} = i_k = 1)$$

As above, direct enumeration is used.

In $5! = 120$ combinations, 54 satisfy the condition.

$$Pr(i_{k-2} = i_k = 1) = \frac{54}{120} = \frac{9}{20}$$

$$\therefore \quad cov(i_{k-2}, i_k) = \frac{9}{20} - \left(\frac{2}{3}\right)^2 = \frac{1}{180}$$

but from (1) and (2) since

$$cov(i_{j-2}, i_j) = \gamma \quad \forall j$$

$$cov(i_{j-1}, i_j) = \beta \quad \forall j$$

$$var(i_j) = \alpha \quad \forall j$$

$$\text{Var(Peak Count)} = (n-2)\alpha + 2(n-3)\beta + 2(n-4)\gamma$$

$$= \frac{2}{9}(n-2) - \frac{1}{18}(n-3) + \frac{1}{90}(n-4)$$

$$= \frac{5n}{18} - \frac{29}{90}$$

So, we have the mean and variance of the peak count statistic, under the hypothesis that {X} is IID. The Central Limit Theorem may be used to show that, if n is sufficiently large, the sum of the peak indicator variates is approximately Normal. This allows hypothesis testing to take place.

CHAPTER 5

A BAYESIAN APPROACH TO CRIME PATTERN ANALYSIS

5.1 A Brief Outline of Bayesianism

The Bayesian interpretation of statistical inference and measurement differs fundamentally from the classical in several ways. The most basic of these differences is in the definition of probability. Classically, the probability of an event is defined in terms of relative frequency (See eg Kyberg and Smokler, 1963). The probability of a particular outcome of an experiment or process is the limit of the proportion of times that this particular outcome occurs as the experiment is repeated indefinitely. An important corollary of this is that classical probabilities are only defined for infinitely repeated events. In Bayesian terms, a probability is defined in terms of "degree of belief". Before an event occurs, it is a measure of the likelihood of particular outcomes occurring. This definition is more generally applicable, and in this framework unique or finitely reproducible events may also have probabilities.

In terms of inference, the Bayesian model combines prior beliefs about some hypothesis with experimental evidence (ie. data) to produce "posterior beliefs". Given the Bayesian definition of probability, the prior and posterior beliefs are specified by probabilities and are related by Bayes' theorem:

$$P(A|X) = \frac{P(X|A)\,P(A)}{P(X|A)\,P(A) + P(X|\bar{A})P(\bar{A})} \qquad (1)$$

where A is the hypothesis
$\bar{A}$ is the negation of the hypothesis
X is the observed data
$P(.)$ is the notation for probability

Here $P(A)$ is the prior belief
$P(A|X)$ is the posterior belief
$P(R|S)$ is the notation for the probability of E given S)

Thus, to perform a Bayesian hypothesis test, a prior belief is needed, together with a probability model for the observed data given the hypothesis and its negation.

Surprisingly, the Bayesian method of parameter estimation is identical to this. If the hypothesis A is now treated is an infinite set hypothesis of the form $\underline{\theta} = \underline{\kappa}$,

$$P(\underline{\theta}|X) = \frac{P(X|\underline{\theta})P(\underline{\theta})\big|_{\underline{\theta}=\underline{\kappa}}}{\sum_{\underline{\theta}} P(X|\underline{\theta})\,P(\underline{\theta})} \quad \text{in the discrete case} \qquad (2)$$

or $$P(\underline{\theta}|X) = \frac{P(X|\underline{\theta})P(\underline{\theta})\big|_{\underline{\theta}=\underline{\kappa}}}{\int_{\underline{\theta}} P(\underline{X}|\underline{\theta})\,P(\underline{\theta})\,d\underline{\theta}} \quad \text{in the continuous case}$$

Here, the probability of a hypothesis is replaced by a probability distribution (or density function) of the parameters.

This approach to data analysis offers several advantages over the classical. Firstly, input of knowledge prior to the experiment is

allowed. This can be an attempt to represent subjective beliefs, or perhaps results from some past experimentation. In the event of no prior knowledge, the concept of a "non-informative prior" is introduced. For example, in the hypothesis testing case $P(A) = P(\bar{A}) = 1/2$ represents equal prior evidence both for and against A. Non-informative prior formulations for parameter estimation are considered in Box and Tiao (1973).

Secondly, there is a conceptually simpler measure of experimental evidence. The Bayesian probability $P(A|x)$ is a direct probability of the hypothesis. A classical significance level for hypothesis testing is a statement about the testing process. It is the classical probability of wrongly rejecting A if A is in fact true, viewed in terms of the probability space of X. Similarly, for parameter estimation, classical confidence limits are defined in terms of probability of containing $\underline{\theta}$, given the sampling probabilities of X, whilst Bayesian analysis provides a distribution for the value of $\underline{\theta}$.

Lastly, sequential testing is more naturally provided for in Bayesian theory. In the hypothesis testing case, given a set of independent observations $\{x_1, \ldots x_n\}$

, at any integer $K < n$, $P(A|x_1 \ldots x_K)$ is simply defined in equation (1). Thus, viewing the experiment as a process evolving in time, a measure of evidence is easily evaluated at each intermediate data collection point. Sequential hypothesis testing in classical analysis is considerably more complicated; see for example (Wald, 1947).

A further feature of this Bayesian property is considered below. At point K,

$$P(A|x_1 \ldots x_K) = \frac{P(A)P(x_1 \ldots x_K|A)}{P(A)P(x_1 \ldots x_K|A) + P(\bar{A})P(x_1 \ldots x_K|\bar{A})}$$

and, at point $K+1$,

$$P(A|x_1 \ldots x_{K+1}) = \frac{P(A)P(x_1 \ldots x_K)P(x_{K+1}|A)}{P(A)P(x_1 \ldots x_K)P(x_{K+1}|A) + P(\bar{A})P(x_1 \ldots x_K)P(x_{K+1}|\bar{A})}$$

$$\vdots$$

$$= \frac{P(A|x_1 \ldots x_K)P(x_{K+1}|A)}{P(A)P(x_1 \ldots x_K)P(x_{K+1}|A) + P(\bar{A})P(x_1 \ldots x_K)P(x_{K+1}|\bar{A})}$$

Thus, the prior of the observation of x_{K+1} is the posterior after x_K. This seems intuitively reasonable. However, in the Bayesian framework, this allows for modification of priors in the instance of extra-experimental evidence, at time $K+1$. Thus, not only may degree of belief be monitored during the experiment, but it may be modified, all within the theoretical framework of Bayesianism. All of the above may also be applied to parameter estimation also, by starting the above mathematical reasoning applying equation (2) at point K.

All of the above may be applied in a predictive context. In this case, the outcome of a future event Y, is considered in terms of known data, X and a parameter being estimated, $\underline{\theta}$. Firstly, $P(\underline{\theta}|X)$ is obtained using (2), and then

$$P(Y|X) = \int_{\underline{\theta}} P(Y, \underline{\theta}|X)\, d\theta$$

$$= \int_{\underline{\theta}} P(y|\underline{\theta}, x)\, P(\underline{\theta}|x)\, d\underline{\theta}$$

$$= \int_{\underline{\theta}} P(y|\underline{\theta})\, P(\underline{\theta}|x)\, d\underline{\theta}$$

(In the continuous case)

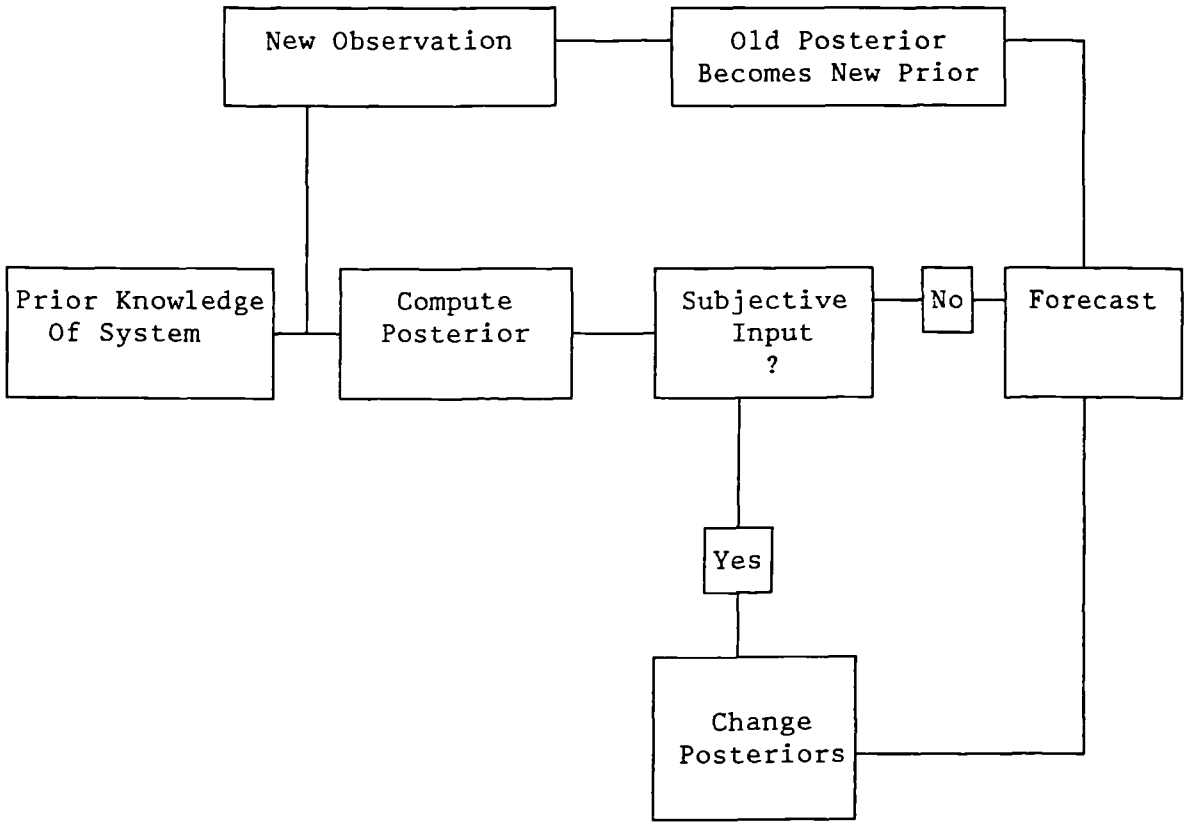
As time evolves, updated versions of $P(\theta|x)$ will be derived, and the predictions will be based on better informed posterior distributions for $\underline{\theta}$. Again, any new subjective information, or results from external attempts to measure $\underline{\theta}$ may be used to update $P(\theta|x)$ at any time. The Bayesian forecasting model may be set out as in Figure 1 (Fyldes, 1984).

5.2 Bayesian Forecasting Applied to Crime Pattern Analysis

In the last section, it was outlined how Bayesian analysis may generally be applied to forecasting, given past data. The methodology will now be considered in the particular context of crime forecasting. Clearly there are features that may be exploited in this situation.

Now, collection of data on household burglaries may be possible, and the analysis of the last chapter showed that, due to certain degree of space-time epidemicity, past crime patterns can often provide strong clues as to future evolution of geographical pattern; but there are other pieces of information which could improve predictions, but which may not easily be incorporated into the formal database. Such information may, for example, consist of the knowledge that a known repetitive offender has returned to an area, or conversely that a criminal active in a certain region has recently been convicted. Data such as this may provide rates may rise or drop, despite differing evidence suggested by spatial

Figure 5.1


New Observation
Old Posterior Becomes New Prior
Prior Knowledge Of System
Compute Posterior
Subjective Input ?
No
Forecast
Yes
Change Posteriors

patterns in past rates. The Bayesian scheme of figure 5.1 incorporates this type of information, and would allow the combination of human prior knowledge with quantitative pattern analysis techniques allowing both aspects of crime pattern analysis to be incorporated into a prediction model.

5.2.1 Spatial Aspects

The results of chapter three may also be incorporated into the scheme of figure 5.1. The expectation of spatial structure within the beatwise rates in each week may be quantitatively expressed in terms of an initial prior distribution for parameters governing rates: ie. as "prior knowledge of the system" in the diagram.

They may also be expressed in the likelihood function of the data. It should be noted at this point that, as in the last chapter, the prediction is applied to a multidimensional system within a subdivision. Thus, autocorrelation structures will be the medium through which perceptions of spatial structure are expressed in the prior distribution.

Typically, the stochastic models proposed in the last chapter may be applied in the Bayesian context, expressing certain degrees of initial belief in the distributional paraders for crime rates at week n, given those for week $n-1$, "as prior knowledge of the system". Clearly, for example one expects $P(\alpha<0)>P(\alpha>0)$ and due to clustering, $P(c>c)>P(c<c)$ in a process described by

$$\mu_i = \sum_{j \in S/i} (d_{ij}^{-\alpha} x_j) + e \qquad x_{i,t-1} = E(x_{i,t}), \quad x_{i,t} \sim N(\mu_i, \sigma^2)$$

5.3 The Human Computer Interface

So far, a forecasting model has been considered in terms of a Bayesian framework, requiring various multivariate probability distributions, as well as data, as an input, and yielding a further multivariate probability distribution as an output. These are reasonable ways of expressing degrees of belief to a user of the system who is familiar with the concept of multivariate distributions, and therefore capable of drawing interpretations from inputs and outputs of the system in its purest form. However, generally one does not expect a target user a system such as this to be familiar with these concepts. It is unlikely that police officers will undergo an intensive training course in Bayesian probability theory in order to use this system!

It seems more reasonable to devise means in which prior subjective beliefs, and posterior crime production distributions may be handled in more familiar formats, and in which some form of interface will convert these formats into the type of information required by the prediction system.

The output distribution function will be considered first. Generally, to summarise a Bayesian distribution, some forms of descriptive statistics will be computed.

Thus, for example, the mean, median or the mode of a distribution may be used for point estimation. In the predictive case, these will provide point forecasts for the coming weeks crime rates. For interval estimates

in the one dimensional case, a Bayesian analogue of confidence intervals is used. An interval (a,b) having certain properties is evaluated, under the condition that $\int_a^b P(\theta|x)d\theta=\alpha$ for some prescribed α. Here α is $P(a<\theta\leq b)$ This does not uniquely specify α, and another condition has to be imposed. Commonly it is specified that $P(\theta<a) = P(\theta>b)=\frac{1}{2}(1-\alpha)$, or alternatively $P(\theta)>P(\phi)$ for all $\theta\in(a,b)$ and all $\phi\notin(a,b)$. These intervals may be extended to regions for the multivariate case.

In the case of point estimates, mapping the predicted values onto a beat map of the police subdivision may be an easily interpreted method of posterior prediction distribution representation. This may be done either using proportional symbol mapping, choropleth mapping or directly labelling beats on a map with the predicted crime rates. A survey to discover which of these is the most successful representation is considered in greater detail in chapter 7.

A problem of dimensionality is encountered when dealing with interval or region estimations. However, some information of this type should perhaps be incorporated into the system. This attaches a measure of certainty to the predictions. Effectively, a beat whose predicted value has a wide interval is more likely to deviate from its predicted value. Also, given that the prediction distribution is multivariate, those beat pairs having high convenience should also be considered. It therefore seems reasonable that the task of a map-based output is to convey the first and second moments of the predictive distribution in a format interpretable by police officer users.

The solution for the first moment has already been considered. A possible solution for the second moment may be to offer two further maps: one highlighting "least predictable beats", the other "highly related beats" if the respective variance and correlation figures become sufficiently large. It may be possible that some users may prefer to ignore the more detailed information, and use the basic forecasted map pattern; however, the second moment related maps may be offered as options, to be studied if the user requires further information.

Next, the input of subjective information will be considered. To some extent, if a human operator is aware of the "epidemic effects", one would expect their predictions to tie in with that of the predictor in the system. It is more important that the user supplies extra information, of courses of crime pattern not detectable in past data of crime rates. One way of gaining this information may be to display the predictions obtainable when only using the posterior distributions from the spate time stochastic model, and asking if there is any way in which the user disagrees with this prediction.

If the space-time pattern analysis, and the operators knowledge agree on predictions, they remain unaltered. However, any knowledge unique to the human analyst may now be given an opportunity to enter the system. As stated before, it would be unreasonable to ask for a probability distribution at this point. A simple menu-based modification of the forecasts will be more simply visualised by the operator. It may also be suggested again that some information as regards second moments could

be input. This may be done by a dialogue between the user and the machine asking

(1) "How variable do you feel this forecast is?"

and (2) "Do you think a change in forecast in this beat would affect any other beats?"

The first of the responses could be used to readjust variances in the covariance matrix, and the second to alter the non-diagonal elements.

5.4 The Introduction of Advanced Bayesian Techniques

In the above sections it has been outlined, a way of adapting a basic Bayesian forecasting scheme for Police use. Some more advanced aspects of Bayesian theory will now be considered, and it will be discussed how these aspects may be used to improve the outlined forecasting system. The advancements will include a formal specification for combining Bayesian priors, in this case formulating a means of combining police operator prior beliefs with those based on analysis of past data. Also, a method of calibrating priors based on past performance of their source will be considered. This introduces a property of "adaptability" into the prediction model: consistently poor forecasters will tend to become "down weighted" in the predictions, whilst good performers will have increasingly greater leverage on predictions.

In addition to the above improvements, the concept of multi-state modelling will be considered. In this approach, the possibility of sudden possibly spurious changes in process is considered. For example, a particular beat may incur a spuriously high number of household burglaries in a given week, but then return to normal, or the arrest of an offender particularly active in an area may cause the overall level in that region to drop suddenly. There are several ways in which Bayesian probabilistic models may cope with such phenomena, and this will also be discussed.

5.4.1 The Calibration of Bayesian Prior Probability Distributions

As discussed previously, the formal method of inputting prior knowledge about the values of some parameter into a Bayesian system is by specification of a probability distribution, but a major difficulty with this approach is that experts in fields not concerning probability may experience difficulty in expressing their uncertainties in probabilistic terms. However, in a Bayesian framework, if some method of assessing the experts' ability to quantify probabilities exists, this may be used to modify his prior distributions.

A means of doing this (due to Morris, 1974) is outlined briefly below:

Suppose, for a series of exchangeable events (see Kyberg and Smokler, 1963), the expert has provided a prior distribution relating to some prediction or parameter. Subsequently the true value became revealed,

in each case. Then, the cumulative probability of each parameter from the prior distributions can be used as a scale-invariant performance indicator. A value close to zero suggests underprediction, and one close

to one, an unnecessarily high prediction. Also, a value of 0.5 would indicate a good prior, whose median was in fact the true value.

It seems reasonable, then, to extend the idea of a performance indicator for a specific prediction to a distribution of performance indicators, applied to all predictions made by the human analyst. When the user specifies a prior for any given event, there are then two distributions to consider: the prior itself, and a distribution related to the general performance of priors supplied by this user. Adopting an algebraic notation for these quantities, we have

$P^*(\theta_0)$ = prior supplied by the user for (in a general context)

$\phi = \int_{-\infty}^{\theta_0} P(\theta)d\theta$ = cumulative probability, given $\theta = \theta_0$ (performance indicator)

$p(\phi)$ = p.d.f of ϕ

$P(\theta)$ = prior supplied for a specific problem

Then $P(\theta|\phi)$ must now be evaluated, (ie. the distribution of θ if the distribution of ϕ is given). Define

$$Q(x) = \int_{-\infty}^{x} P(\theta)d\theta$$

Then, as Q must be a monotone increasing function,

$$P(\theta \leq \theta_0 \mid \varphi) = P(\varphi \leq Q(\theta_0)) = \int_0^{Q(\theta_0)} \varphi(\phi)\, d\phi$$

Now, the calibrated distribution, $P^*(\theta)$ is the derivative of the above expression with respect to θ.

$$\int_{-\infty}^{\theta_0} P^*(\theta)\, d\theta = \int_{\theta}^{Q(\theta_0)} \varphi(\phi)\, d\phi$$

$$\Rightarrow \quad P^*(\theta) = \frac{d}{d\theta}\left[\int_0^{Q(\theta_0)} \varphi(\phi)\, d\phi\right] = \varphi(Q(\theta_0)) \frac{d}{d\theta} Q(\theta_0)$$

$$= \varphi(Q(\theta_0))\, P(\theta_0)$$

Thus, the eventual calibration is of the form $P(\theta_0) C(\theta_0)$ where $C(\theta_0)$ is a calibration term, defined by $\varphi\left(\int_{-\infty}^{\theta_0} P(\theta)\, d\theta\right)$, ie a function of cumulative probability of θ. Thus, given a probability density function capable of describing the performance of the user, a "correction factor" may be added to the priors that the user supplies.

It is interesting to note that the descriptive powers of the pdf are fairly versatile. Morris (1977) gives examples of curves for indicating overstatement of precision of knowledge, and understatement of this. In addition to this, curves may be given to represent consistent under- or over-estimation.

5.4.2 Estimation of the Performance Distribution

So far, a means of recalibrating a distribution has been given in terms of the performance indicator distribution. However, the task of evaluating such a distribution has not yet been considered. In order to keep strictly to the Bayesian definition of this distribution, it should be evaluated in terms of performance indicators applied to a set of mutually unrelated incidents. In practice, this type of calibration is difficult. In terms of Police officers in the context here, the time and resource overheads lost in performing some sort of experiment to do this may well be prohibitively large. A compromise will have to be reached, where calibration is actually performed on the week-by-week priors given by police officers for their input into the crime prediction Bayesian scheme. This data will be input anyhow, so no extra resource costs will have to be incurred. This also gives an opportunity for an adaptive system.

The accumulation of information about the shape of φ will evolve as a process in time. Initially, nothing will be known about φ, after a few weeks, a fuzzy φ may have evolved: after some time quite an accurate estimate for φ may have been built up. However, it is possible that φ may itself change with time. The most obvious reason for this may the replacement of the main system user with a new operator, the nature of whose prediction abilities differs from the first user. In this case, a method of estimating φ may be able to adapt to a new shape of curve if the observed values suddenly appear to behave differently from the current estimate of φ : for example, in a weekly sample, a goodness of fit test to be carried out between the observed ϕ for each beat and φ. This technique could be flawed, however, as ϕ values in a small geographical region over a single week are not likely to be independent.

Empirically, this could be countered by raising the threshold of the deviance from fit of the model. Generally, if observations are correlated, likelihood of deviation is increased, as a few spurious cases may affect others, which would not occur in an independent model.

Another modification may also be proposed; instead of having a single φ distribution function extend the concept to one of spatial variation: for each beat allow a separate φ_i. This would be equivalent to a model in which the human predictors performance in prior specification varies in space. This is a reasonable assumption: it is possible that, as a police officer, the user may be particularly familiar with some beats in the subdivision, and be a more competent forecaster for these beats than for others.

In this case, the goodness-of-fit, monitoring for fundamental changes in could not be carried out. However, some form of exponential smoothing technique might be applied to the φ estimates, diminishing the effect of values from the distant past. This loss of "fast adaptive response" may possibly be more than outweighed by incorporating a geographical dimension into the performance indicator distribution.

Another important advantage of this type of system is that it is effectively evaluating the priors supplied, rather than the user in person. In a system such as this, wherein prior probability distributions have to be synthesised, a certain amount of unreliability will be introduced into the prior by the synthesis. However, the method proposed here will calibrate the prior in terms of <u>all</u> unreliability,

including that due to the synthesis process. Thus, with enough training data, the system will be able to correct for any design compromises that have to be made in the prior specification routine.

5.4.3 The Combination of Bayesian Prior Distributions

A means of specifying the input prior beliefs of a human user has been discussed in the last section. In addition to this information, there are the prior distributions brought about by the statistical analysis of household burglaries. Thus, some way of integrating these two sources of information becomes necessary, in order to make forecasts, based on both of these factors.

The problem may be tackled with a further application of Bayes' Theorems (Morris 1974, 1977).

It may be seen that, from the position of the forecasting system as a whole, there is a set of beliefs native to the system about the coming week's crime rate, and also a set of beliefs from the external human monitor. In addition to this, native to this system is an observation-based set of beliefs about the external user's performance. Write these as

$p_m(\theta)$ = system prior to θ

$p(\phi)$ = performance distribution for external user

$p_e(\theta)$ = external's uncalibrated prior for θ

For any given θ , say θ_o ,Bayes theorems gives

$$P(\theta_o | P_e(\theta_o)) = K\, P_m(P_e(\theta_o)) P_e(\theta_o) P_m(\theta_o)$$

But, this is just multiplying the assessment of the system by the recalibrated prior. Thus, combination of the two sets of beliefs is a simple multiplicative operation, so long as the nature system has an assessment of the external user's performance in specifying priors.

So far, the case of only one external expert has been considered. However, certain scenarios may occur in which several experts may be entering subjective information. In this case, it would be necessary to specify a method of combining several external priors.

If their beliefs are independent, it can be shown that the multiplicative effect can be logically extended; giving an overall distribution of the form

$$\prod_{i=1}^{n} P_i(\theta) C_i(\theta)$$

for n opinions, represented by $P_i(\theta)$ and calibrated by $C_i(\theta)$. However, the independence assumption is unlikely to be true in practice, particularly in the police user situation. Possibly several officers using the system would discuss recent criminal events in the subdivision, and influence each others views. In this case, the multivariate distribution for performance indicators could not be expressed as a product of individual distributions, but would have to reflect the correlations between performances of the external users.

In this case, it may be shown (Morris 1977) that the prior is of the form

$$C(\theta)\, f_1(\theta) \cdots f_n(\theta)$$

where C is a joint calibration function. If $p(\underline{x})$ is the joint probability density function of the performance vector, then $C(\theta) = p(F(\underline{\theta}))$ where $\underline{\theta}$ is the vector of cumulative prior distributions, ie.

$$F_i(\theta_0) = \int_{-\infty}^{\theta_0} f_i(\theta)\, d\theta$$

This leads to much greater difficulty in estimating the calibration function. Two main problems are then incurred.

(1) The estimation itself becomes more complex, as a multivariate distribution must now be estimated.

(2) There will be a resultant loss in accuracy of estimation. If several users are inputting data, then for a given amount of data, or a given number of weekly predictions, each individual assessment would be based on fewer points. The situation is worsened by the fact that in addition to estimating each individual performance on a relative lack of information, the interrelation between performances must also be measured.

It seems more feasible then, for the input of knowledge to come from a single crime pattern analyst, rather than a set of several officers. It is possible, of course, that different officers might alternatively use the

input facilities. If this were to happen, the calibration applied would not reflect performance of a single human predictor, but of a process involving several officers generating a single prior. In this case, the prior could be recalibrated. However, it is expected that variability in performance of a multi-user generated prior could considerably exceed that of a single user prior, particularly if some users have strongly contrasting views to others. The net result of this would be a general downweighting of the subjective input. For example, if one user had a tendency to over-estimate and another to under-estimate, the system could compensate either of these if they provided sole input. However, mixing together the two users would lead, from the systems viewpoint to an erratic predictor. This would lead to a flat φ function and so to a flat $\mathbb{C}$ multiplying almost by a constant. Unless the system had information as to which user supplied the prior, it would be unsure whether to compensate upwardly or downwardly, and be more likely to virtually ignore input. Given this, it is recommended that the system be defined for a single user.

5.5 Detection of Changes of State and Atypical Phenomena

A further aspect applied to Bayesian modelling may now be considered. The likelihood part of the posterior specification is assumed known for the data used in the forecasting technique. There are times, however when data may deviate notably from this model. As discussed earlier this could occur spuriously, for a single week, or might occur on a long term basis.

In the first case, the effect is something similar to an "outlier". In the second case, it may suggest a more fundamental change in the stochastic model of the process.

An example of the second case could be that, in a particular beat, some houses are demolished. If these were particularly prone to household burglaries (perhaps not being very secure, or having poor protection from intruders in a context of defensible space eg Newman 1972) then their removal may lead to a drop in the average household burglary rate in that beat. If the mean levels for the beat were prescribed from past data analysis, predictions after the point of demolition would be biased above the true rates, and the bias would remain unless the model were re-specified.

Two ways of monitoring for changes in the model are presented here. The first of these allows for the possibility of other models than that most commonly applied to be in force occasionally. The second monitors

performance of posterior beliefs in terms of "surprise". , A surprising result is effectively one to which little probability was assigned to by the prior. The first of these to be considered will be the multi-state model:

5.5.2 Multi-State Model

In this model, there are several probabilistic processes that could generate the data: there is the most usual one, which is used in the prediction process as the normal likelihood function. There is also another possibility, in which a spurious high or low rate is observed. This will be identical to the first model, except that its variance will be very much larger. There is also then a third model, in which other parameters change, and which will subsequently remain changed. The three possibilities are shown for a simple distribution about a mean value in figure 5.2. Refer to these models as M_1, M_2 and M_3 and suppose there may be prior probabilities attached to each of these, as to which is most likely to occur. One would expect M_1 to apply most often, and very occasionally M_2 or M_3 would apply. A reasonable set of probabilities may be

$$P(M_1) = 0.94$$
$$P(M_2) = P(M_3) = 0.03$$

In true Bayesian fashion, on seeing a particular item of data, these probabilities will be modified accordingly. According to Bayes theorem

$$P(M_i|X) = \frac{P(X|M_i)\,P(M_i)}{\sum_i P(X|M_i)\,P(M_i)}$$

Figure 5.2a : Normal Data Pattern

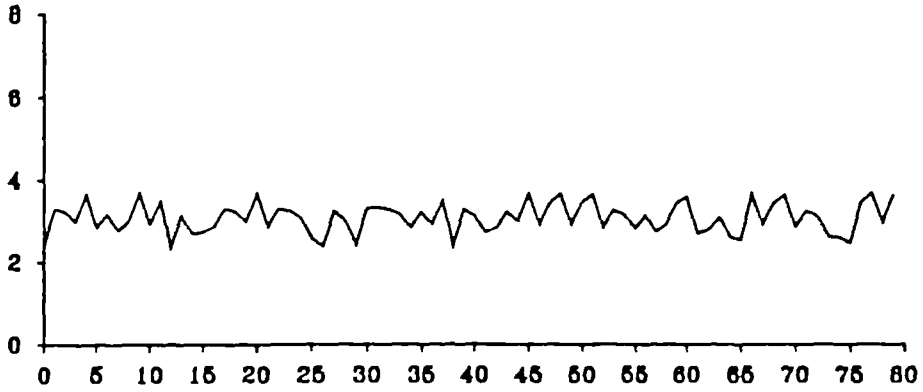


Figure 5.2b : Data With Transient Noise

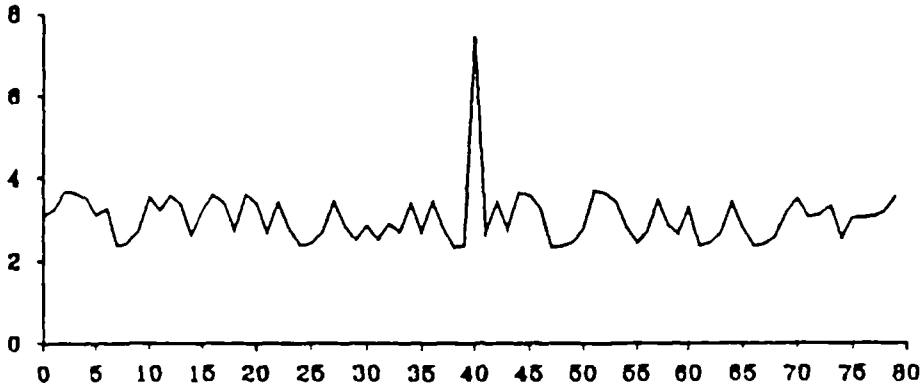
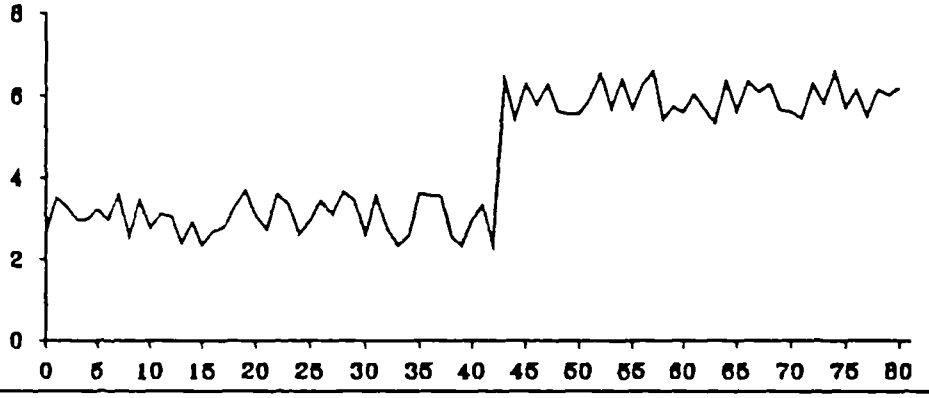


Figure 5.2c : Data With Permanent Shift



where x is the observed data. There are then two alternatives:

(1) From the set of possible models choose the one most likely to have applied in the light of the data observation. Base forecasting in the most likely model.

(2) Base forecasts on the information relating to all of the models, including the probability that each one has occurred.

A method based on (2) is given by Harrison & Stevens (1971). Unfortunately, the prior obtained in this way is a "mixture" of several distributions, and in turn, after two stages we obtain a "mixture of mixtures" and so on. To avoid this, their approach approximates the distribution at each stage by a normal approximation agree to the first and second moments. The author feels that the extra work put into developing a more accurate model by incorporating relative likelihoods of each model may be lost by the last approximation. Perhaps it may be more parsimonious to adopt the first approach.

5.3.3 Atypically Monitoring

An alternative method of checking model performance is outlined here. In this method, the posterior probability prediction distribution in a Bayesian sense, is thought of as a representation of belief as to what will be the outcome of a future event. When the actual outcome is

known, a surprising result could be thought of as one for which the probability in this distribution was low. For example a constant value and could be chosen, and an outcome x defined as "surprising" if $P(x)<\alpha$ It may be more informative, though, to define surprise in terms of the probabilities of x values. Thus result x, is "more surprising" than X_2 if $P(X_1)<P(X_2)$. From this premise, an "index of atypicality" can be constructed, as

$$I = \int_{P(X_1)<P(X_2)} P(X_2)\,dX_2$$

That is, I is the probability of getting an outcome at least as surprising as X. In the case of symmetric, unimodal distributions the value of I is the sum of the upper and lower tail probabilities of $P(X)$.

Here, surprise may be defined as the event $I<\alpha$. Thus, this may be used to flag a spurious event. Repeated surprise may be used to test whether the model is in need of recalibration.

The methodology behind the decision process for this type of modelling is more ad hoc, perhaps, in its approach to identifying changes in the structure of the likelihood model. Despite this, it has other advantages: it only required the specification and calibration of a single model, and also does not require the supply of relative probabilities of the differing models.

This problem accelerates in the multivariate case. In the atypicality index method , modification is fairly simple conceptually. The direct extension

would be to define the atypicality index relating to the entire subdivision as

$$I(\underline{\theta}^*) = \int_{\underline{\theta}:P(\underline{\theta})<P(\underline{\theta}^*)} P(\underline{\theta})\, d\underline{\theta}$$

However, identification of the region over which the integral is to be performed may prove complicated, and beyond this, evaluation of a multi-dimensional integral would have to be performed. An alternative is proposed here, in which the atypicality of each beat in turn is considered: define

$$I(\theta_i) = \int_{\theta:P(\theta)<P(\theta_i)} P(\theta_i \mid \theta_{s/i})\, d\theta_i$$

that is, consider

the conditional distribution of θ_i given the other observed values of θ. A value of θ_i close to that of its neighbours even if high, would not necessarily by surprising. However, if the mean level of the beat i suddenly altered, given that it is deviations about the mean that are considered as correlated in the likelihood models specified in chapter 4, one would expect to get repeated surprising results for that specific beat, given its neighbours values. If the system is then called to intervene, the offending beat mean may undergo recalibration.

In the multi-state model, however, dimensionality brings great complications. It is now possible that each beat may have been in any of the three states. Considering "compound models" to be models in which the states of each beat are considered as a single model, there are 3 possible compound models, each of which would require a prior

probability assignment. This could be simplified by assuming the states to occur independently in each beat (ie. probability of state i in beat j is independent of states of any other beats) but in a spatial process this is unlikely. Spurious high rates in neighbouring beats are fairly likely to have affect across boundary lines. In the atypicality model, unless there is a sudden change across several beats all having common boundaries, a certain amount of conditional atypicality would be observed. Suppose, for example the dark shaded beats in figure 5.3 had surprisingly high rates. Although the adjacency of, say 1 & 2 may reduce the conditional surprise index slightly, the effects of (3, 9, 8) on 1 and (5, 6, 7, 8) on 2 should still make conditional surprise fairly high.

Thus, in the multivariate case, as in here, a mechanism for determining spurious high or local rates, or when the model may need to be recalibrated, would be most practically based on atypicality monitoring.

5.5.4 Practical Example

In this section, to help evaluate the practical aspects of both methods, they are compared for a simple univariate example. This should help to illustrate how the principles discussed in the last section are put into practice. In implementing a one dimensional model, and facing some fundamental problems concerning the method generally, it is hoped that the more sophisticated multi-dimensional system may be approached with greater initial understanding. It also gives a means of comparing the atypicality monitoring and more theoretically sound multi-state models, to

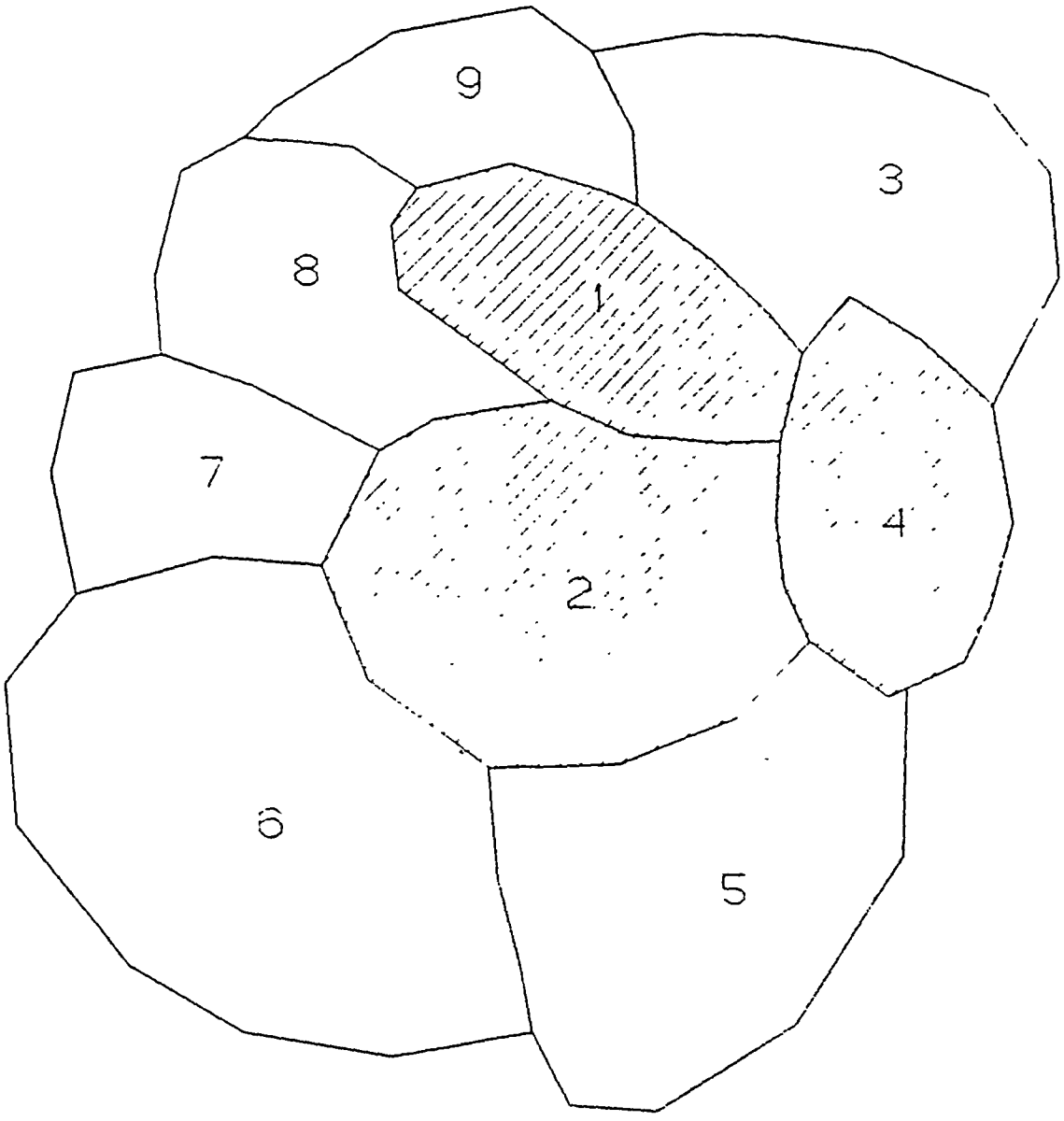


Figure 5.3

assess any losses possibly incurred by adopting the former of the multidimensional model.

The Problem

The problem here is simplistic. A process is defined as normally distributed white noise about a mean μ_a, having variance σ_a^2. However, occasionally there is a spurious observation having mean μ_a but variance σ_b^2. Also, occasionally, a jump occurs in the system, and μ_a is replaced by $\mu_a + \delta$, where $\delta \sim N(0, \sigma_c)$.

The task here is to evaluate μ_a, and identify points in time when spurious high variance noise occurs, and points in time at which a jump occurs. At this point, the previous estimate of μ_a should be discarded.

The process is actually defined thus

$$\mu_a = 1.0$$
$$\sigma_a = 0.5$$
$$\sigma_b = 4.0$$
$$\sigma_c = 3.0$$

State space

P (normal observation) = 0.90

p (high variance) = 0.05

p (jump) = 0.05

5.5.4.1 Multi-State Approach

Assume that the probabilities for each state is given. Then the likelihood, given an observation $\underline{X}$, for the normal state is given by

$$\frac{0.9\,NDF(\mu_a,\sigma_a)}{0.9\,NDF(\mu_a,\sigma_a)+0.05\,NDF(\mu_a,\sigma_b)+0.05\,NDF(\mu_a+\delta,\sigma_c)}$$

likelihood for the alternative state are given by

$$\frac{0.05\,NDF(\mu_a,\sigma_b)}{0.9\,NDF(\mu_a,\sigma_a)+0.05\,NDF(\mu_a,\sigma_b)+0.05\,NDF(\mu_a+\delta,\sigma_c)}$$

and

$$\frac{0.05\,NDF(\mu_a+\delta,\sigma_c)}{0.9\,NDF(\mu_a,\sigma_a)+0.05\,NDF(\mu_a,\sigma_b)+0.05\,NDF(\mu_a+\delta,\sigma_c)}$$

if it is wished to decide which of the three models generated $\underline{x}$, this should be done by selecting the one giving the largest value in (3).

These estimations however, assume perfect knowledge of $\sigma_a, \sigma_b, \sigma_c$ and μ_a. It is more likely however, that some of these will be unknown, or be expressed as priors. In this case,

$$P(X|M_i)$$

should be replaced with

$$P(X|M_i, \mu_a \cdots)\, P(\mu_a)$$

Suppose in this case that reasonable knowledge from past data exists for the σ values, but not for μ_a.

Then the prior for μ_a has to be built up as knowledge of $\underline{x}$ increases. This is done in the usual way, by multiplying the prior by the likelihood

of the event, and normalising. However, although the true likelihood of the event is the mixture,

$$\sum P(M_i)\, P(x \mid M_i, \mu_a)$$

, to avoid complication, the most probable state likelihood will be considered as the likelihood function. If it is suspected that a change of state has occurred, however, the prior for μ_a will be reset to the non-informative prior. This is actually an "improper prior" (Barnett, 1982). This will also be the prior initially.

At each stage of reading in an x value, an estimate of μ_a will be output. This will be in terms of the mean of the current posterior for (incidentally as this will be a normal distribution, the estimate could identically be defined as either the median or mode of the distribution). In addition to this, it will be flagged if either the jump or spurious state is thought to have occurred.

5.5.4.2 Atypically Index Solution

Again, it is assumed that variances are given, but not the mean value Thus, a prior distribution for μ_a must be supplied. In the initial state, no prior knowledge about μ_a exists, so the constant non-informative prior is assumed. After the first observation, the posterior in μ_a is

$$NDF(\mu_a, \sigma_a)$$

So the posterior is normal, with a mean of $\bar{x}$. This represents belief as to the next likely value. A surprising result occurs in t, when x_{t+1} is revealed next time, if

$$P(x_2|x_1) < \alpha \quad \text{for some } \alpha$$

$$\text{ie} \quad \int e^{-\frac{1}{2}\left(\frac{x_2-\mu_a}{\sigma_a}\right)^2} e^{-\frac{1}{2}\left(\frac{x_1-\mu_a}{\sigma_a}\right)^2} d\mu_a < \alpha$$

This is also based on a normal distribution since this distribution is symmetric and unimodal, a surprising result $P(x_2|x_1)$, is equivalent to a result $x_2 < K_1$ or $K_2 > x_2$ where K_1 and K_2 are the upper and lower $\alpha/2$ tails of the distribution.

In this case, choose surprise at 5%. Then it is necessary to monitor for

$$x_{n+1} \geq K_2(x_1 \ldots x_n)$$

$$x_{n+1} \leq K_1(x_1 \ldots x_n)$$

If this occurs once, it is first considered to be a spurious result. If, however, the surprise recurs (in the same tail) this may give the impression that a jump has occurred.

In the instance that an initial spurious result is thought to have occurred, the posterior belief is not modified (as it is not thought that the likelihood function generally used to model applies in this case). If a second "surprising" result occurs, again the result is not used to modify the prior. On the third instance, it is assumed that a jump has

occurred. Then, the prior distribution for is reset to the non-informative prior, and re-calibration begins.

5.5.5 Results of Simulation

The system described in the last section was simulated using usual methods of random number generation for Gaussian variates (see eg Newman and Odell, 1971). The points at which a random spurious observation occurred, and the points at which a jump occurred were noted, together with the actual series generated. This data was then fed into the two algorithms proposed in the last section. Their performance is shown in figures 5.5 (multi-state model) and 5.6 (surprise model). The simulated series is shown in figure 5.4. At each incidence of a spurious high variance observation, both methods were capable of flagging the event. The multi-state model flagged most cases of spurious variation, although often flagged jump-states as spurious variation.

In the surprise model, there was no facility to flag jumps immediately. However, the "repeated surprise" parameter appeared more effective in some ways. Although the multi-state model was capable of rejecting the "normal" model, it was often unable to identify jumps, erroneously flagging a spurious observation. This often happened on several consecutive data item inputs. It seems that the time-dimension in jump detection, although empirical here, plays an important role.

Figure 5.4 : Simulated Data Series

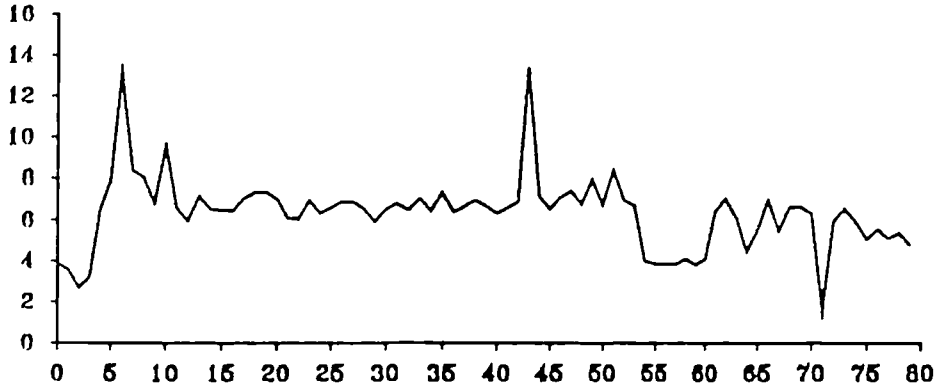


Figure 5.5 : Multi-State Predictor

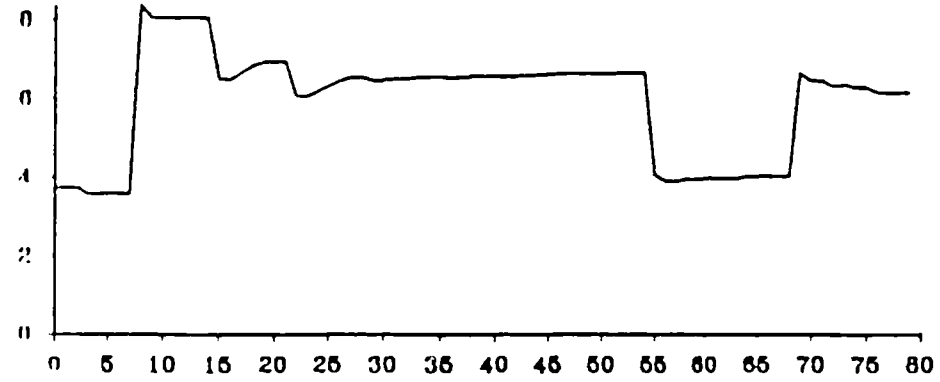
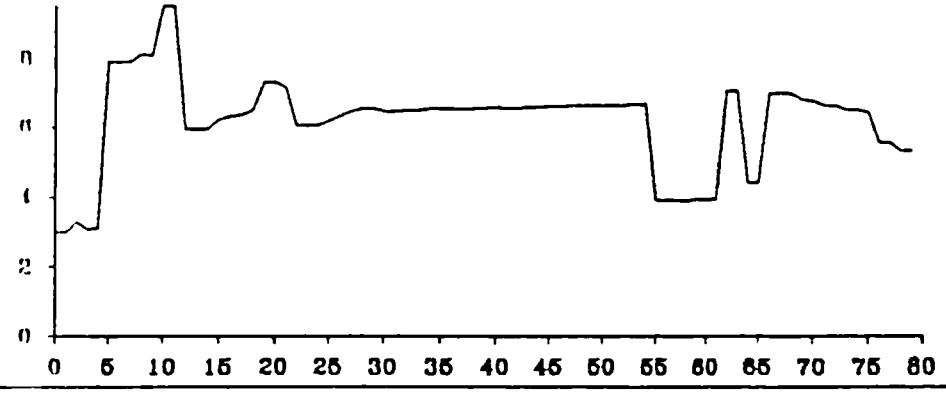


Figure 5.6 : Surprise Index Predictor



In this simulation, then, both models were relatively effective, although time dimension considerations made the "surprise" model more effective at jump detection.

Another important point to consider is that whilst the multi-state model exploited information about alternative states to the usual, in the form of alternative probability models, the surprise model did not require this, but was able to detect jumps (albeit fairly large ones) using information about the main distribution alone.

It also may be of importance to consider how capable of detecting relatively minor jumps both of these systems are. In each case, a certain amount of re-calibration may be required. For the multi-state case, new details about the "jump parameters" would have to be input. For the "surprise" model, and could be lowered to allow the possibility of jumps to be detected with more sensitivity .

If, as is likely to be the case in order to constrain computing overheads, tests for "jumps" or "spurious effects" are likely to be performed independently between beats, it appears that the conceptually simpler surprise test would be the most applicable. Certainly, from the results of this simulation, they seem to perform similarly, with, if anything, slightly more powerful results from the "surprise" method.

5.6 Review of System Design

At this stage of the study, the typical Bayesian forecasting system suggested in 5.1 could now be revised. In the past sections, a means of calibrating input priors has been proposed, as a means of combining the priors of the forecasting system, obtained from data analysis, and those of the human user.

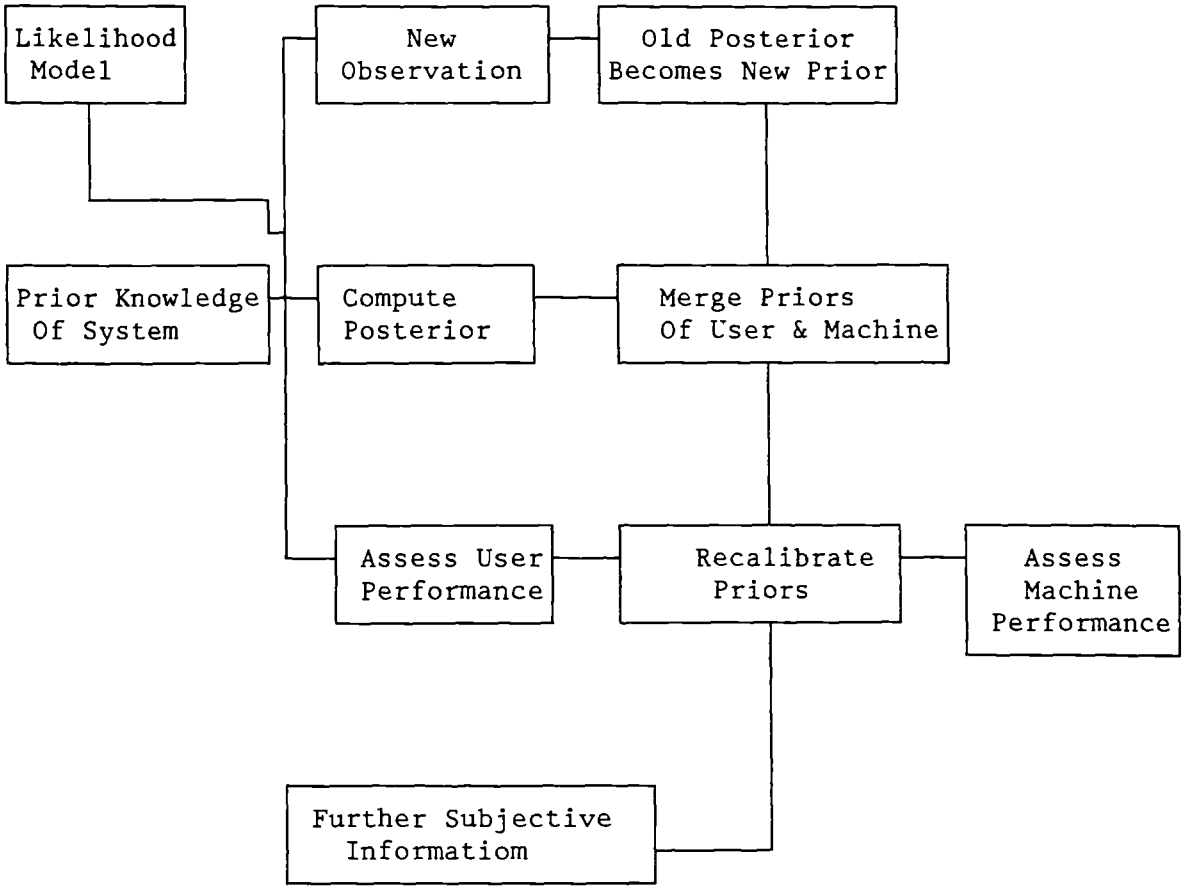
Finally, some thought has been given to detection of sudden changes in the likelihood model, and how these may be incorporated into a prediction system. The possibility of several human users giving input has also been considered, although this was not eventually recommended.

The incorporation of these extra techniques seems reasonable, so that the scheme may now be expanded, giving the revised system in figure 5.7. It is this scheme that is to be proposed for the model of a prediction system in the prototype. In the next section, development of the individual parts of the system will be considered in detail.

5.7.1 Application of Bayesian Analysis Techniques to a Computer Crime Forecasting System

In this chapter so far, various techniques of Bayesian analysis have been considered. Attention has also been given to multivariate spatial aspects of these techniques. These will now be combined in a specific context, based on the analysis of household burglary incidents as a space time process given in chapter 4. The forecasting system can then

Figure 5.7


Likelihood Model
New Observation
Old Posterior Becomes New Prior
Prior Knowledge Of System
Compute Posterior
Merge Priors Of User & Machine
Assess User Performance
Recalibrate Priors
Assess Machine Performance
Further Subjective Informatiom

be thought of as a multivariate prediction method combining prior knowledge of the spatial and temporal structure of the process, evidence from beatwise data collected on household burglaries from the past, and human expert knowledge of beats at risk, due to circumstances undetectable in data. The design of the system will be based on the diagram in figure 5.7. Each aspect of this will now be considered in term.

5.7.2. A Space-Time Series Model

As evident in Chapter 3, the process may be stochastically modelled as a space-time process. Results implied that, given a vector of household burglary densities suitably transformed by a square root transformation, $\underline{X}_t$ could be modelled as being related to $\underline{X}_{t+1}$ using a space-time autoregressive formula.

$$\underline{X}_{t+1} - M = A(\underline{X}_{t+1} - M) + B(\underline{X}_t - \underline{M}) + \underline{\varepsilon}$$

$$\underline{\varepsilon} \sim N(0, \sigma)$$

Thus, $\underline{X}_{t+1}$ could be modelled as a random vector, whose conditional probability on $\underline{X}_t$ could be expressed as

$$P(X_{t+1} | \underline{X}_t) = K \exp\{-(\underline{X}_{t+1} - \underline{M}^*)^T \Lambda (\underline{X}_{t+1} - \underline{M}^*)/2\}$$

$$\underline{M}^* = \underline{M} + A(\underline{X}_t - \underline{M})$$

Clearly, if $\underline{X}_{t+1}$ is conditionally independent on all $\underline{X}_{t-K+1}$, $K>1$, as suggested in chapter 4 (there was little improvement in the model when a term in $\underline{X}_{t-2}$ was entered), then the series $\{\underline{X}_t\}$ has a Markov property:

$$P(\underline{X}_{t+1} | \underline{X}_t, \underline{X}_{t-1} \cdots \underline{X}_{t-K}) = P(\underline{X}_{t+1} | \underline{X}_t)$$

Given a series of weekly household burglary vectors, $\underline{X}_1 .. \underline{X}_t$ we have

$$P(\underline{X}_1 .. \underline{X}_t) = P(\underline{X}_t | \underline{X}_{t-1} .. \underline{X}_1) P(\underline{X}_{t-1} | \underline{X}_{t-2} .. \underline{X}_1) \cdots P(\underline{X}_1)$$
$$= P(\underline{X}_t | \underline{X}_{t-1}) P(\underline{X}_{t-1} | \underline{X}_{t-2}) \cdots P(\underline{X}_1)$$

If $\underline{X}_1$, is taken as fixed

$$P(\underline{X}_2 \cdots \underline{X}_t | \underline{X}_1) = P(\underline{X}_t | \underline{X}_{t-1}) P(\underline{X}_{t-1} | \underline{X}_{t-2}) \cdots P(\underline{X}_2 | \underline{X}_1)$$

Thus, if $\underline{X}_1$ is known (which it will be), then the likelihood of the remaining data will just be the product of the conditional likelihood if each weeks observation, given the previous weeks.

Thus

$$P(\{\underline{X}_n\} | \underline{X}_1) = \prod_{i=2}^{t} K \exp\{-(\underline{X}_i - \underline{\mu}^*)^T \Lambda (X_i - \underline{\mu}^*)/2\}$$
$$= K^{2/n} \exp(-\tfrac{1}{2} L)$$

where
$$L = \sum_{i=2}^{t} (\underline{X}_i - \underline{\mu}^*)^T \Lambda (\underline{X}_i - \underline{\mu}^*)$$
$$= \sum_{i=2}^{t} (\underline{X}_i - A\underline{X}_{i-1} + (A-I)\underline{\mu})^T \Lambda (\underline{X}_i - A\underline{X}_{i-1} + (A-I)\underline{\mu})$$

(expanding)
$$= (t-1)\left(\underline{V} - \frac{\Sigma \underline{X}_t}{t-1} + A\frac{\Sigma \underline{X}_t}{t-1}\right)^T \Lambda \left(\underline{V} - \frac{\Sigma \underline{X}_t}{t-1} + A\frac{\Sigma \underline{X}_{t-1}}{t-1}\right)$$
$+$ terms excluding $\underline{V}$ $\quad (\underline{V} = (A-I)\underline{\mu})$

If t is sufficiently large, we have $\sum_{2}^{n} \underline{X}_t \approx \sum_{1}^{n-1} \underline{X}_t = \hat{\underline{\mu}}$, say

$$\therefore \quad L = (t-1)(\underline{V} - (I-A)\hat{\underline{\mu}})^T \Lambda (\underline{V} - (I-A)\hat{\underline{\mu}})$$
$$= (t-1)(\underline{\mu} - \hat{\underline{\mu}})^T \Lambda^* (\underline{\mu} - \hat{\underline{\mu}}) \qquad (\Lambda^* = (I-A)\Lambda(I-A))$$

If it is only necessary to estimate $\underline{\mu}$, so that A and Λ are assumed known, and initially adopt a non-informative prior for $\underline{\mu}$, then the above expression is proportional to the posterior for $\underline{\mu}$, given the data set. Thus: $\underline{\mu} \sim N(\hat{\underline{\mu}}, \Lambda^{*-1}/(t-1))$ This is an intuitively sensible expression. If $\underline{\mu}$ remains fixed, and $t \to \infty$, then the variance tends to zero, and $\hat{\underline{\mu}}$, the running mean vector of $\{\underline{x}_t\}$ tends to $\underline{\mu}$.

If predictions were to be based on this posterior distribution, then the density of $\underline{x}_{t+1}$ would be given by

$$P(\underline{x}_{t+1} | \{\underline{x}_t\}) = \int_{\mathbb{R}^n} P(\underline{x}_{t+1} | \underline{\mu}) P(\underline{\mu} | \{\underline{x}_t\}) \, d\underline{\mu}$$

Now $$P(\underline{x}_{t+1} | \underline{\mu}) P(\underline{\mu} | \{\underline{x}_t\}) = \exp(-L/2)$$

where
$$L = (t-1)(\underline{\mu} - \hat{\underline{\mu}})^T \Lambda^* (\underline{\mu} - \hat{\underline{\mu}}) + ((\underline{x}_{t+1} - \underline{\mu}) - A(\underline{x}_{t+1} - \underline{\mu}))^T \Lambda ((\underline{x}_{t+1} - \underline{\mu}) - A(\underline{x}_{t+1} - \underline{\mu}))$$

$$= t(\underline{V} - \tfrac{t-1}{t}\underline{V} - \tfrac{1}{t}(\underline{x}_{t+1} - A\underline{x}_t))^T \Lambda (\underline{V} - \tfrac{t-1}{t}\underline{V} - \tfrac{1}{t}(\underline{x}_{t+1} - A\underline{x}_t)) + (t-1)\underline{V}^T \Lambda \underline{V} + (\underline{x}_{t+1} - A\underline{x}_t)^T \Lambda (\underline{x}_{t+1} - A\underline{x}_t) - \tfrac{(t-1)^2}{t} \hat{\underline{V}}^T \Lambda \hat{\underline{V}} - 2\tfrac{t-1}{t}(\underline{x}_{t-1} - A\underline{x}_t)^T \Lambda \hat{\underline{V}} - \tfrac{1}{t}(\underline{x}_{t+1} - A\underline{x}_t)^T (\underline{x}_{t+1} - A\underline{x}_t)$$

integrating out the expression in $\underline{V} = (A - I)\underline{\mu}$, we have

$$P(\underline{x}_{t+1} | \{\underline{x}_t\}) = \exp(-L'/2)$$

where

$$L' = (1 - \tfrac{1}{t})(\underline{x}_{t+1} - A\underline{x}_t)^T \Lambda (\underline{x}_{t+1} - A\underline{x}_t) - 2(1 - \tfrac{1}{t})(\underline{x}_{t+1} - A\underline{x}_t)^T \Lambda \hat{\underline{V}} + \text{terms not including } \underline{x}_{t+1}$$

$$= (1 - \tfrac{1}{t})(\underline{x}_{t+1} - A\underline{x}_t - \hat{\underline{V}})^T \Lambda (\underline{x}_{t+1} - A\underline{x}_t - \hat{\underline{V}}) + \text{terms excluding } \underline{x}_{t+1}$$

$$= (1-\tfrac{1}{t})(\underline{X}_{t+1} - A(\underline{X}_t - \hat{\underline{\mu}}) - \hat{\underline{\mu}})^T \Lambda (\underline{X}_{t+1} - A(\underline{X}_t - \hat{\underline{\mu}}) - \hat{\underline{\mu}})$$

Thus $$\underline{X}_{t+1} \mid \{\underline{X}_t\} \sim N(\hat{\underline{\mu}} + A(\underline{X}_t - \hat{\underline{\mu}}), (1-\tfrac{1}{t})^{-1}\Lambda^{-1})$$

Again, this is intuitively appealing if $\underline{\mu}$ is fixed, then as $t \to \infty$, $(1-\frac{1}{t})^{-1} \to 1$ from above, and $\hat{\underline{\mu}} \to \underline{\mu}$, so that, asymptotically

$\underline{X}_{t+1} \mid \{\underline{X}_t\}$, the predictive distribution for $\underline{X}_t$, tends to the stochastic model for $\underline{X}_{t+1} \mid \underline{X}_t$, when Λ is known.

In particular, a point prediction for burglary rates at week $\underline{X}_{t+1}$ can be obtained from $A(\underline{X}_t - \hat{\underline{\mu}}) + \hat{\underline{\mu}}$, where $\hat{\underline{\mu}}$ is the vector of running mean rates. This, however, assumes that A is a fixed quantity. Also, if confidence limits are required, Λ is required also, and again, at this stage Λ is a fixed quantity.

It is possible to incorporate estimators of A and Λ into the Bayesian modelling system, thus returning a posterior distribution of the form:

$f(A, \Lambda, \underline{\mu} \mid \{\underline{X}_t\})$ This, however, would be problematic if all of the elements of and were to be estimated, albeit in a symmetric format, there would now be, for an n-beat system, $n + n(n-1) = n^2$ variable parameters to estimate, as opposed to n. There are also not all normally distributed. We have, for example,

$$P(\Lambda, A, \underline{\mu} \mid \{\underline{X}_t\}) = K|\Lambda|^n \prod_i \exp\left(\tfrac{1}{2}(\underline{X}_i - \underline{\mu} - (A\underline{X}_{t-1} - \underline{\mu})^T \Lambda (\underline{X}_i - \underline{\mu} - (A\underline{X}_{t-1} - \underline{\mu})\right)$$

is non-normal in Λ. Also for predictive distributions an integral over Λ and A is now required. The dimensional complexity may be reduced on

the basis of results in chapter 4. A uniform autoregression coefficient for all neighbouring beats will provide good results in relation to allowing each coefficient to vary. In the case of models A-E from section 5 in chapter 4, it could be put

$$A = a_1 A^*$$

where $A^*_{ij} = 0$ if beats are not adjacent and

d_{ij}^{-1} if they are.

In this case, the only unknown parameter is a_1. Allowing also for regression effects of the same beat on the previous week, then

$$A = a_0 I + a_1 A^*$$

This reduces the number of parameters to two for A.

similarly, put $\Lambda = b_0 I + b_1 A^*$

Then, there are only $n+4$ parameters.

Consider first the instance where a_0 is known as are b_0 and b_1. Then we have

$$P(a_0, \underline{M} \mid \{x_i\}, a_1, b_0, b_1) \propto \exp(-L)$$

where

$$L = (t-1)\left(\underline{V} - \frac{\Sigma \underline{X}_t}{t-1} + A\frac{\Sigma \underline{X}_{t-1}}{t-1}\right)^T \Lambda \left(\underline{V} - \frac{\Sigma \underline{X}_t}{t-1} + A\frac{\Sigma \underline{X}_{t-1}}{t-1}\right)$$
$$- 2\frac{\hat{\underline{M}}^T \Lambda A \hat{\underline{M}}_0}{t-1} + \frac{\hat{\underline{M}}_0 A \Lambda A \hat{\underline{M}}_0}{t-1} + \sum_{i=2}^{t} \underline{X}_{t-1}^T A^T \Lambda A \underline{X}_{t-1}$$
$$- 2\sum_{i=2}^{t} \underline{X}_{t-1}^T A^T \Lambda \underline{X}_t$$

$$\hat{\underline{\mu}}_0 = \frac{\sum_{i=2}^{t} \underline{x}_{i-1}}{t-1} \qquad \hat{\underline{\mu}}_1 = \frac{\sum_{i=2}^{t} \underline{x}_i}{t-1}$$

and, the marginal distribution for a_0, if $A = a_0 A^*$ is obtained by integrating out the terms of $\underline{V}$, giving

$$a_0 \sim N\left(\frac{\sum (\underline{x}_{i-1} - \hat{\underline{\mu}}_0)^T A^{*T} \Lambda (\underline{x}_i - \hat{\underline{\mu}}_1)}{\sum (\underline{x}_{i-1} - \hat{\underline{\mu}}_0)^T A^{*T} \Lambda A^* (\underline{x}_i - \hat{\underline{\mu}}_1)}, \frac{1}{(t-1)\sum (\underline{x}_i - \hat{\underline{\mu}}_0)^T A^{*T} \Sigma A^* (\underline{x}_i - \underline{\mu}_0)} \right)$$

incorporating a_1 yields similar results, with a bivariate normal distribution in a_0 and a_1.

However, the predictive distribution for $\underline{X}$, becomes considerably more complex.

The predictive distribution is no longer normal. Asymptotically, however, this will be the case: it can be shown (Box and Tiao, 1973) that $P(\underline{x}_{t+1} \mid \{\underline{x}_t\}) \to P(\underline{x}_{t+1} \mid \mu, a)$ as $t \to \infty$. Thus, the distribution should tend to normality if $\{\underline{x}_t\}$ is a sufficiently large sample.

However, problems will still be encountered when employing models in which is not diagonal. In combining the predictive distribution for $\underline{x}_t$ with that of the user prior, the resultant distribution will be of the form:

$$k\, f(\underline{x}_{t+1})\, e^{-(\underline{x}_{t+1} - \underline{\mu}) \Lambda (\underline{x}_{t+1} - \underline{\mu})/2}$$

Suppose, for example, $f(X_{t+1})$ is a normal function, then the combined distribution is a multivariate normal, with variance - covariance matrix $(\Lambda^{-1}+\Lambda_u^{-1})^{-1}$, and mean vector $(\Lambda^{-1}+\Lambda_u^{-1})^{-1}(\Lambda_u \underline{X}_{t+1} + \Lambda \underline{X}_{t+1}^{u})$

where the u denotes parameters for the user's distribution. Thus, a matrix conversion will be necessary; in the case of a subdivision of n beats, a new conversion will be necessary. When n is typically between 30 and 40 beats, this will lead to computational difficulties on currently available micros. Thus, a compromise must be reached, in which Λ is a diagonal matrix. This is equivalent to a space-time series in which, although the effect of adjacent beats is considered at a lag effect, the expected deviations from the predicted values are modelled as being independent. This corresponds to models A, B, and C in section 4 of chapter 4. The best performing model in this set is C, so this will be adopted.

Finally, again considering computational simplicity, the coefficients and could be estimated from a training data set. may still be estimated using "live" data. It is possible, that if atypicality monitoring, as proposed in section , is employed, then after several "deviant" predictions are obtained, the spatial model could be re-calibrated using a new training data set.

5.7.3 Morris Type Calibration of User Predictions

Having discussed Morris's method of re-calibrated user supplied prior distributions, a means to implement this numerically must now be

proposed. As discussed, it is necessary to obtain a "performance function" which is a probability distribution of the cumulative probability of events occurring, as specified by the users prior. This can be calibrated once outcomes (in this case crime counts) are known.

In this model, the square roots of crime counts are approximately normally distributed (see chapter 4, section 2), so that normal priors will combine with these to give normal posterior distributions. Thus, if the user supplies a mean and a standard deviation for each beat i; (μ_i, σ_i) the cumulative probability of obtaining a rate (after square root transformation) is $\Phi(\frac{x-\mu_i}{\sigma_i})$ where Φ is the cumulative normal distribution function

$$\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx$$

This may be obtained using Hastings' approximation (Ambramowitz and Stegun, 1972).

From a set of such Φ-values for each beat, a performance function can be built up. This may be done in several ways. Firstly, the method of Kernel estimators could be used (see chapter 4, Section 2 , or chapter 3 section 2). This builds up a model of a distribution of a variable X, given an observed set of x values $\{x_1 \ldots x_n\}$.

However, this may be inapropriate here, since it gives equal weighting to all observed Φ values. A more adaptive technique would give a higher weighting to recent values; it is possible that a users

performance may vary with time. Therefore, a method of modelling the performance function which "forgets" results in the distant past, based for example on exponential smoothing, is proposed. For example, if each observation's contribution is defined as a Beta distribution on $(0,1)$, then the performance function F may be estimated as

$$F_t(p) = \alpha F_{t-1}(p) + (1-\alpha)(1-p)^{1-a} p^a / \beta(a, 1-a)$$

where α is chosen to maximise $(1-x)^{1-a} x^a$ at $\Phi\left(\frac{x-\mu_i}{\sigma_i}\right)$

and $\beta(a, 1-a) = \int_0^1 (1-x)^{1-a} x^a \, dx$

A problem with this type of estimator, however, is that even if Φ repeatedly takes the same value, the variance of $F_t(p)$ will not decrease, since $F_t(p)$ will tend to $(1-p)^{1-a} p^a / \beta(a, 1-a)$. Thus, another solution may be a multiplicative estimator.

$$F_t(p) = K F_{t-1}(p) \left[(1-p)^{1-a} p^a\right]^{\alpha}$$

K is chosen to normalise $F_t(p)$ α is used to determine the rate at which the variance decreases if similar performances occur repeatedly.

Here, if Φ repeated takes the same value,

$F_t(p) \to \delta(p - \Phi)$ where δ is the Dirac delta function (see, eg Wiley and Barrett, 1982 or Queen, 1980).

Having obtained an estimation for $F_t(p)$, it is now necessary to obtain the corrected predictive distribution for the user, from that supplied.

Again if it is assumed that the distribution is normal, then the corrected distribution is

$$\frac{1}{\sqrt{2\pi\sigma^2}} F_t\left(\bar{\Phi}\left(\frac{x-\mu_p}{\sigma_p}\right)\right) e^{-\frac{1}{2}\left(\frac{x-\mu_p}{\sigma_p}\right)^2}$$

For prediction purposes, for each beat the mean and variance of this distribution is required. These are defined by the integrals

(1) $$\text{Mean} = \int_{-\infty}^{\infty} \frac{x}{\sqrt{2\pi\sigma_p^2}} F_t\left(\bar{\Phi}\left(\frac{x-\mu_p}{\sigma_p}\right)\right) e^{-\frac{1}{2}\left(\frac{x-\mu_p}{\sigma_p}\right)^2} dx$$

(2) $$\text{Variance} = \int_{-\infty}^{\infty} \frac{x^2}{\sqrt{2\pi\sigma_p^2}} F_t\left(\bar{\Phi}\left(\frac{x-\mu_p}{\sigma_p}\right)\right) e^{-\frac{1}{2}\left(\frac{x-\mu_p}{\sigma_p}\right)^2} dx - (\text{mean})^2$$

These may be evaluated using a Gauss - Hermite Formula (Atkinson, 1978).

These give approximations of the form

$$\int_{-\infty}^{\infty} f(x) e^{-x^2} dx \simeq \sum_{i=1}^{n} \omega_i f(x_i)$$

in formulae (1) and (2), put $y = \left(\frac{x-\mu}{\sqrt{2}\sigma_p}\right)$

Then mean $= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} (\mu_p + \sqrt{2}\sigma_p y) F_t(\bar{\Phi}(\sqrt{2}y)) e^{-y^2} dy$

and variance $= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} (\mu_p + \sqrt{2}\sigma_p y)^2 F_t(\bar{\Phi}(\sqrt{2}y)) e^{-y^2} dy$

Clearly, if $F_t(\bar{\Phi}(\sqrt{2}y))$ can be evaluated, these expressions are of the correct form for Gauss-Hermite approximation.

If F_t is stored as an array of point estimates at regularly spaced intervals, then, using interpolation techniques, $F_t(p)$ can be approximated for arbitrary p-values. In this case, for the mean, the approximation formula is

$$\frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} w_i (\mu_p + x_i \sqrt{2}\sigma_p) f(x_i)$$

where $f(x_i) = F_t(\Phi(\sqrt{2}x_i))$. A similar formula may be obtained for the variance. Thus, a mean and variance for a modified user prior can be approximated. In the initial state, put $F_0(p) = 1$. In this case, in the light of no experience of the users predictive performance, the users prior remains unmodified. Thus, the algorithm for the users' prior is

(1) Initialise $F_t(p)$ to 1 at all points

(2) Read prediction

(3) Modify prediction using $F_t(p)$
approximate mean and S.D. using Gauss-Hermite.

(4) Read actual crime count

(5) Modify $F_t(p)$ by evaluating Φ from actual crime count and user's prediction.

(6) Return to step 2.

These are incorporated into the listing of the prototype system. A simple trial of the multiplicative and exponential smoothing methods is shown in figure 5.8. The solid line indicates a simulated series with a deliberate jump included, and the dotted lines indicate upper and lower 5%

Figure 58b: Multiplicative Model Correction

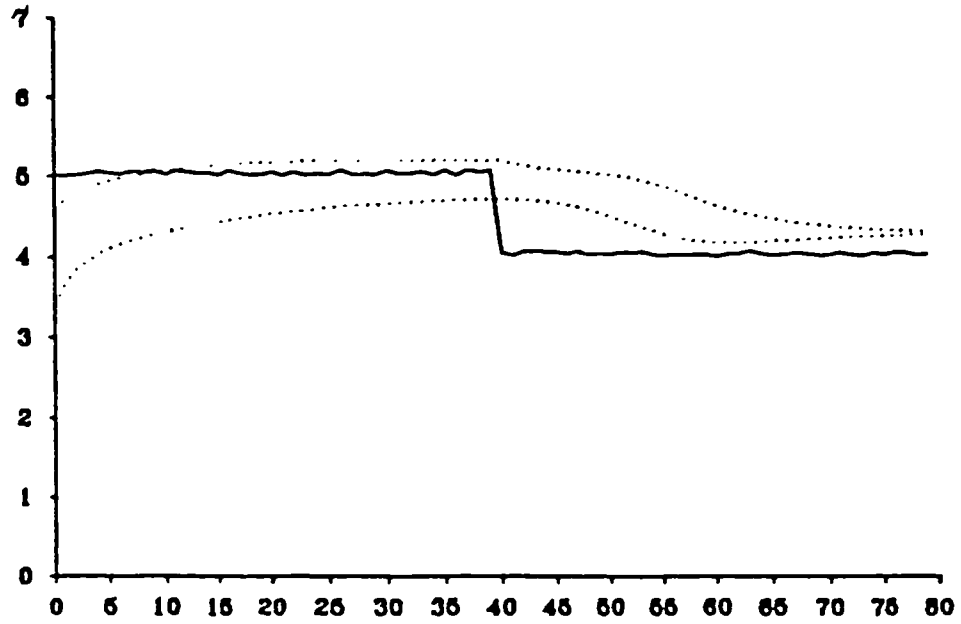
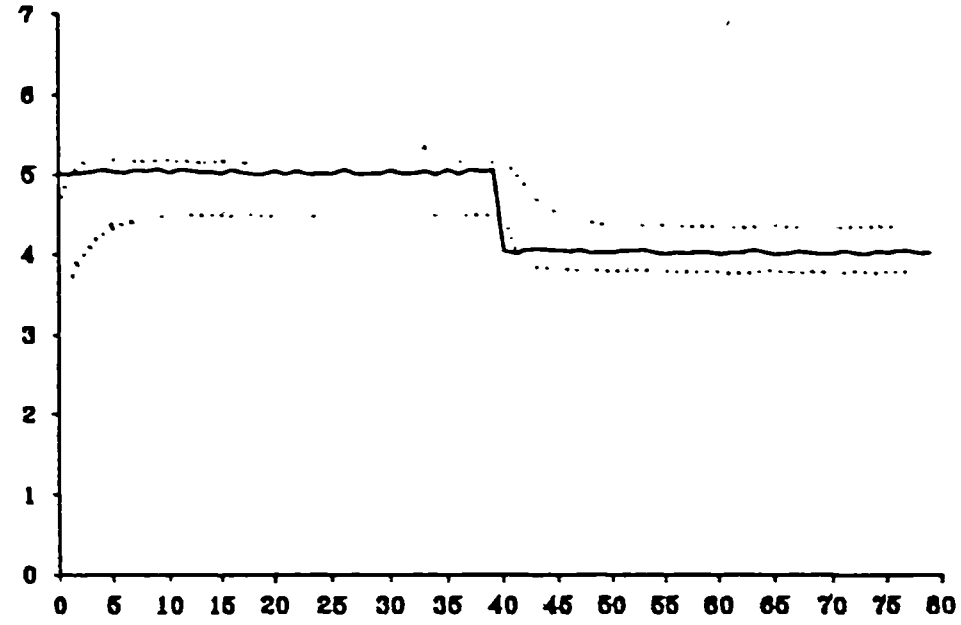


Figure 58a: Exponentially Smoothed Correction



credibility bounds. The performance of the exponential smoothing model seems more desirable than that of the multiplicative model, since although at times the exponential model gives a smaller credibility interval, bias introduced when a state change occurs in the model effects the mean predicted level adversely for an undesirably long period.

5.7.4 Combining User and Machine Predictions

At this state, machine predictions using space-time autoregressive models are available, as are user predictions. The machine predictions for future rates of crimes $\underline{X}_{t}$ based on $\underline{X}_{t}$, are given by

$$E(\underline{X}_{t+1}) = \hat{\underline{\mu}} + A(\underline{X}_{t} - \underline{\mu}) \qquad Var(\underline{X}_{t+1}) = ((1 - \tfrac{1}{t})\Gamma^{-1}\Lambda^{-1})$$

where $\hat{\underline{\mu}}$ is a running mean estimate for $\underline{\mu}$, over t time periods. If Λ is assumed diagonal, then the prediction can be interpreted as a set of independent priors for each beat, with mean μ_i, and variance $\lambda_{ii}^{-1}(1-\frac{1}{t})^{-1}$. If atypicality correction is employed, the variance would become $\lambda^{-1}_{ii}(1-1/t_{ii})^{-1}$ where t_i is the number of time periods to over which $\hat{\mu}_i$ has been estimated , since the last change in value was implemented.

Using the estimates of mean and variance for the corrected user priors for each beat, an overall mean, variance pair may be computed. Multiplying the two normal probability functions together gives a further normal distribution, with mean

$$\frac{\sigma_m^2 \mu_u + \sigma_a^2 \mu_p}{\sigma_m^2 + \sigma_u^2}$$

and variance $\left(\sigma_m^{-2} + \sigma_u^{-2}\right)^{-1}$

(see, for example Barnett 1982).

where the suffix m denotes parameters supplied by machine,and u for user. Thus, there is a weighted mean combination of both predictions which may be used to obtain an overall prediction.

If the variances are used as a measure of "confidence" then the weighting favours the most "confident" forecast.

5.8 Conclusions

In this chapter, the principles of Bayesian inference have been outlined, and in particular applied to certain problems of forecasting. This approach has been extended to the calibration of user's prior belief specifications, in order to correct tendency to over or under predict, and also to allow for these tendencies to vary spatially. Provision has then been made to incorporate this type of prediction with a forecast based on past data patterns as laid out in chapter 4. Finally, the problem was applied to the specific problem of crime prediction using a micro. Although current micro technology may restrict some or the more complex space-time autoregressive models, an effective model from chapter 4 has been implemented. This allows an adaptive, self-calibrating prediction model, incorporating the spatial and temporal nature of the crime data to be implemented on a micro, for eventual use in police subdivisions.

CHAPTER 6

THE IMPLEMENTATION OF A BAYESIAN CRIME PREDICTION SYSTEM ON A MICROCOMPUTER

6.1 Introduction

Having chosen a Bayesian approach to crime prediction, and identified the needs of a crime prediction system to be used by police forces, it now follows to operationalise these results by implementing a Bayesian prediction system on a micro, to be used on site. The aim of this chapter is to set about this task, paying close attention to the ease of use of such a system. In a Bayesian crime prediction package, there is a need for a database to be built up, and also for the subjective beliefs of expert police users to be input in some way, resulting in a prior probability distribution. If the 'man-machine interface' in such a system is poor, not only would there be an increased chance of entering incorrect crime reports into the database, but also incorrect prior belief representations may result. Thus, carefully worded and easily corrected requests for input from users are essential for the reliable running of the system.

Thus, in this chapter, design of an informative, user friendly software system will be attempted. Also important is the method of extracting prior beliefs from operators, to produce Bayesian prior probability distributions. Clearly, it is not reasonable to expect the operator to specify an algebraic representation of their prior distribution outright,

thus methods for building prior distributions by asking the operator to specify levels of crime risk in a local geographical sense will also be investigated.

The above paragraphs refer to the design of the software for a Bayesian crime prediction systems. In addition to this, this chapter aims to choose a hardware configuration, and realise the algorithms attained in the design section in some programming language. Thus, the ultimate aim is to create a working crime prediction system, which may then be used for "on site" testing of a crime prediction system in subdivisions of a Police Force.

6.2 Design Specifications of Program

Since the program will be required to offer several options to the operator, some control of the program at run time must be offered. This could be done either with a command language or by displaying several screens of menus, and asking the user to make selections from these. In this case the menu driven approach will be adopted. This may be justified since menu driven software has been found to be used more efficiently by non-expert users, and even for expert users, this may minimise the number of keystrokes required to access different parts of the system (Savage and Habinek, 1984).

The menu-based input/output routines should be easily modifiable so that future extension of the system can easily be carried out. Ideally, the operating system of the computer should be accessible from within a

program. Thus, a particular section of the system could be selected from menus, and the menu-calling program could then initiate the program to perform the selected task. The menu program could be controlled by a 'menu control file', consisting of the text for each of the menu items, a program name to be run if selected (or possibly another menu to be called up), and possibly some help screens.

The idea of such a system is that if new features were to be incorporated in the future, this could be done by writing a new program, and editing the control files without breaking into existing software. In fact, new parts of the system could be implemented in a different language to the existing software since each feature would be a separate module held together by the operating system, accessed from the menu program.

The means of communicating between these programs will be achieved through standard format data files. The main data required by the system will be records of past crimes (which must be updated regularly) digitised boundaries for maps, used in graphical display of past data, and information relating to the performance of the users' prior beliefs, which are necessary in a Bayesian forecasting set-up. While the information about past crimes and performance of prior beliefs will be dynamic, changing with time, the boundaries of police beats for map drawing are more static. A higher level of admittance to data editing would be required to modify these files.

Also, some form of security must be implemented in this system. Confidential information is being processed, and furthermore only those with permission should be allowed to enter data into the system. Thus, certain parts of the system should be protected by asking the operator for a password. Again, this password may be stored in the menu program control file, but to preserve confidentiality, should be in encrypted form. As with the digitised map updating software, the setting of passwords should be done by a user with a higher level of access rather than an everyday user.

Certain programs will offer graphics facilities: maps of the subdivision, highlighting risk areas, showing beatwise crime predictions and so on. The derivation of these programs will be covered in the sections concerning graphics design. Finally, other programs will carry out the mathematics required to make predictions (using Bayesian analysis). The design of software concerning the input and output of information will be discussed here, but not that of performing the analysis. This software follows naturally from the chapter concerned with developing the final prediction model to be used. The development of the system here is merely a direct translation from the mathematical formulae arrived at in the last chapter. The input software for Bayesian prior beliefs must be considered carefully, however, as must the software for output of past data as they may all be considered as links in the chain of communicating knowledge to the operator, and feeding his reaction to this back to the Bayesian inferential computation process. Serious flaws in either aspect will lead to poor performance in the predictor.

Having considered the design requirements of the system, it now follows to take each of the above aspects in turn and create algorithms to attain the specified objectives.

As outlined above, tasks performed by the software are best controlled via a system of menus. These are used more efficiently by inexperienced users (possibly foot beat constables who only denote a small proportion of their time to entering crimes into the database). The way this is to be achieved is by having a 'father' program, which displays menus on the VDU, which will initiate other 'child' programs residing on disk when these are chosen from the menu. When these are running, the 'father' program is frozen, to re-start when execution of the 'child' has terminated.

When running, the program will require information about the menus to be displayed. Principally, it will need to know the text describing each option on the menu, together with instructions on what to do if a choice is made. Also, since security will be important, information as to whether the instruction may only be carried out on correct entry of a password, together with the password, may be stored. Finally, a line of explanatory text, stating in simple terms the action carried out by each menu choice, will improve 'user-friendliness' and hopefully reduce the frequency of errors made in the system. All of these pieces of information will be required for every choice on the menu. In addition to this, a menu header, giving a title to the set of choices being offered (ie. 'Future Crime Rate Prediction Display'), only one of which is required per menu, could be added.

A possible framework for achieving this is a virtual 'operating environment' that sits inside the main operating system. If possible, this environment could be initiated when the micro is switched on, as part of a starting-up procedure. A flow diagram of the control program is given in figure 6.1

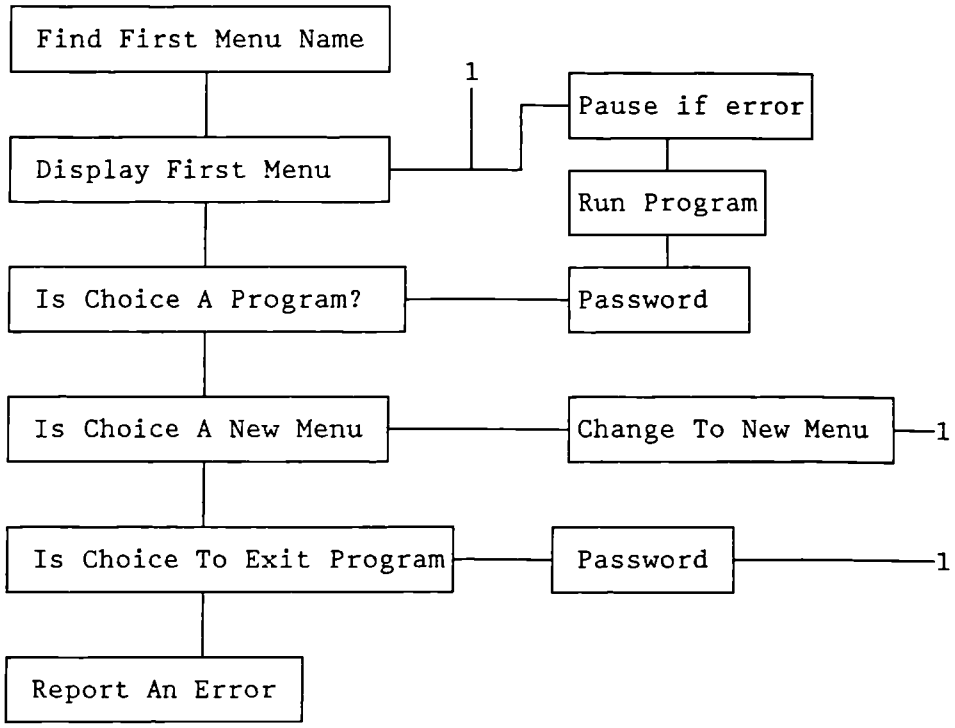
A suggested layout for the menu is given in figure 6.2. Full use is made of the screen area, and choices on the menu are double spaced. Provision is given for a menu title, and room for an explanatory line for each option is provided. This layout implies guidelines for the maximum length of titles, and menu choice text lines.

6.2.1 Modules for the Control Program

In the past sections, a "control program" has been specified. This program has the ability to call other programs written to perform specific tasks. It now follows to consider the options that should be offered in order to specify the collection of programs to be accessible in the prototype system. Clearly, to operationalise the Bayesian system proposed in chapter 4, it is necessary to include a prediction module, allowing analysis of data and incorporation of the subjective advice of police officers. This module will also allow results of Bayesian prediction analysis to be displayed on the VDU. It is convenient to keep both of these tasks in the same module:- part of the user prediction input depends on the display of data from the machine prediction, so that it is convenient to switch between both of these without leaving the module.

Figure 6.1

Flow Diagram Of Control Program



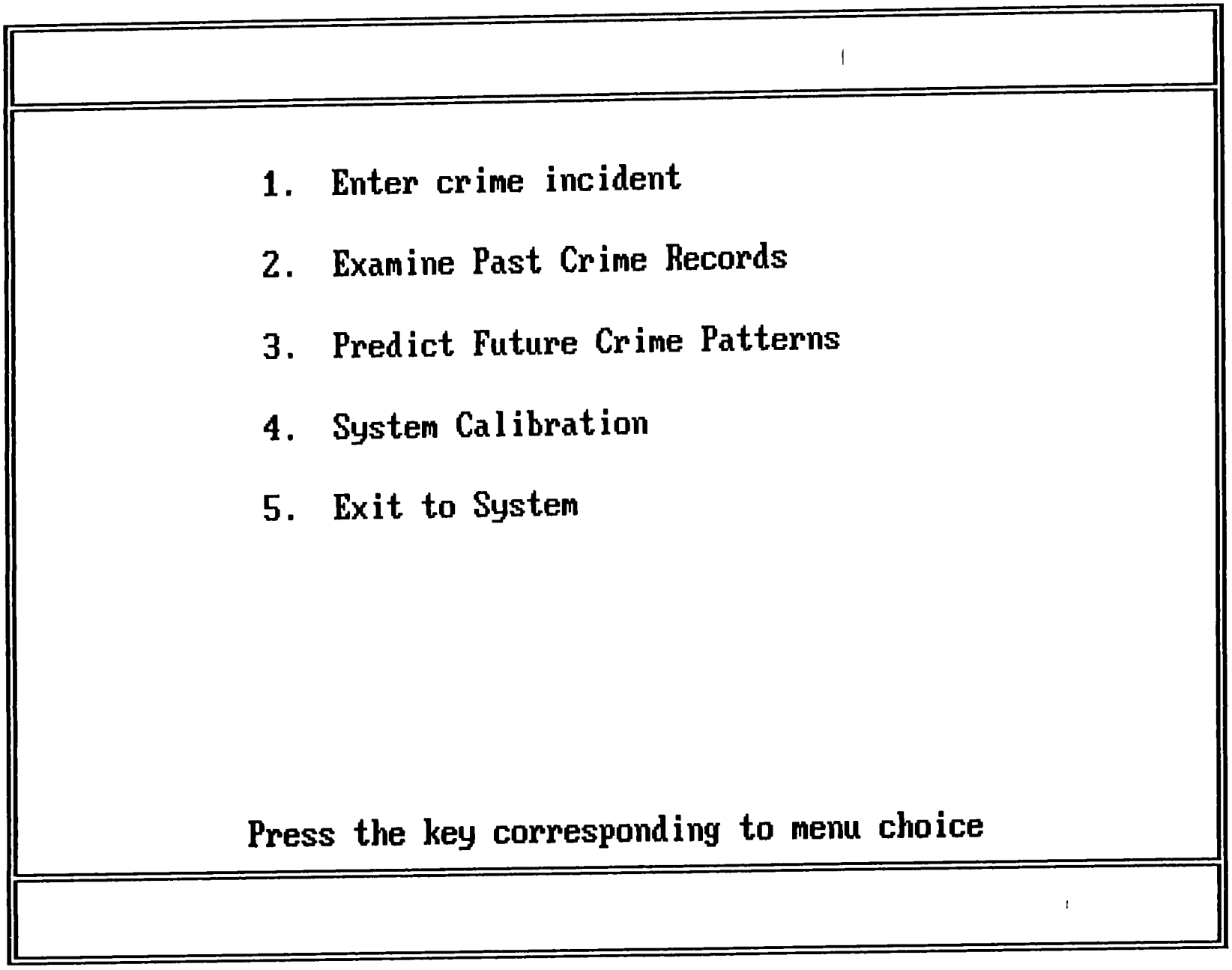

1. Enter crime incident
2. Examine Past Crime Records
3. Predict Future Crime Patterns
4. System Calibration
5. Exit to System
Press the key corresponding to menu choice

Figure 6.2

Also, for this module and any other analysis module, there needs to be some means of inputting data. Thus, a module dedicated to management of the database needs to be created. This module should be capable of reading in new data items, and expanding database files accordingly. These database files will then be used by other modules. As with the control program a user friendly data input and error checking system is important to ensure thorough and reliable data input. It is possible at some point in the future that this module may be replaced by a communications module, which will be able to read data from a force-wide database in the central headquarters, acting as a file server to several crime prediction and analysis systems of various subdivisions. However, although a certain amount of crime detail is currently centrally recorded within the Northumbria police force, insufficient geographical detail within subdivisions is stored, and networking software capable of transmitting data for this type of analysis is not present on the central system. This implies that, at least on a prototype system, local database building techniques must be used, although at some future point they may be superseded.

This serves as an example of the modular philosophy behind the design of this system. Provided a communications based data reader maintains the local database in exactly the same format as the local data input system, it is only required to alter menu descriptor files, and remove the old data program, replacing it with the new. No access to the control software, or any other modules, will be necessary.

Arising from chapter 4, it is notable that many methods of mapping the past data help to shed some light on spatial processes taking place. It therefore seems important to include methods of mapping past data into other modules. A beatwise choropleth mapping module would be useful, for example, when considering the allocation of resources to beats. Also, if past records of "surprisingly high" beats, as set out in chapter 4, are kept, these may also be mapped. It would be advantageous here, as with the menu systems, if the display was coloured. Differing intensities of one colour could provide the basic choropleth information (say light blue) and a different colour used to highlight the "surprising beats".

In addition to beatwise mapping, which may be useful to resource management, it would be helpful to officers operating on specific beats to give pointwise information of past data. Indeed, the knox tests given in chapter 4, in a mapped form, may provide useful analytical output to identify locations of burglary "epidemics" at a sub-footbeat level.

In addition to providing greater detail, this type of mapping may identify "crime clusters" straddled across beat borders. The aggregated choropleth maps, however, may fail to identify such phenomena (Openshaw, 1984).

Finally, another section of chapter 4 dealt with kernel estimation (Silverman, 1978 a or b). Whilst Knox testing (Knox, 1964) identifies clusters in space and time for burglaries, the approach for kernel estimation works purely in a spatial sense. A "risk surface" is

constructed over the region served by the subdivisional force officers, the height of which signifies the risk per unit area of household burglary.

The surface may then be contoured, and those points of highest attitude shaded, and projected onto a map of the subdivision. This then gives a map of "high risk" regions for household burglaries. Clearly, the technique may eventually be applied to any spatially referenced crime incident (eg. violent assault, public disorder, etc). The map may be periodically revised. Again this gives information in terms of manpower deployment, as well as presenting spatial aspects of crime risks. High risk regions overlapping beat boundaries may be identified, so officers on adjacent beats including the same high risk regions may be notified. If, eventually, risk surfaces for all combined crime incidents are generated, they may provide useful evidence for the reorganisation of beat boundaries. Thus, this type of mapping of past data together with choropleth and point mapping all have useful applications which merit their inclusion into the system as modules. Also, it seems useful at times to be able to output the cross-tabulated beatwise data in text form. This conveys the same information as the choropleth maps, but in a non-spatial format. However, in the format of the printed page the information may be photocopied and circulated around the subdivision.

Finally, in the initial analysis of the data, some other techniques were considered, for example to analyse seasonal aspects of the data (Chapter 2) or to examine relative risk to household burglaries at various times of day (also Chapter 2). Some of these techniques may be reliant on a

database other than the main database, and, at least in the prototype stage, difficult to integrate fully into the system. However, if separate, static databases are maintained to service these routines, they may at least be run from the control program, although at present updating may be done by conventional file handling techniques within the operating system. These modules may appear peripheral to the main system, but some of the techniques discussed and developed in the earlier part of this thesis, although not central, may be useful for certain aspects of crime pattern analysis. It therefore seems reasonable to include them in some form within a prototype system, to allow their assessment in an operational environment.

It is possible that this set of modules could exist on a separate menu, distinguishing them from the "mainstream" features, and when implementing the prototype make clear that these features are at a considerably less developed stage than some of the other features, in terms of their database management. If it is speculated after evaluation that any of these may provide useful information, further development will be justified. If these features were excluded from a prototype, opportunities of identifying new areas of statistical and mapping crime pattern analysis software may be missed.

6.2.2 Equipment Configuration

In this section, a microcomputer system on which to implement the system will be proposed. Up until now, although consideration has been given to the specification of methodology for the forecasting and display

of geographical crime patterns, little thought has been given to the practicalities of implementing such methodologies in a working environment. Clearly, a particular model of microcomputer has to be programmed to perform the specified tasks, and choice of an appropriate programming language and machine is of considerable importance. Although the choice of programming language may be invisible to the end user (a police officer operating the resultant software), this is of importance when the package is being developed. Any language used must allow access to data of the type required by the prediction methods, flexible means of interactive communication with the user, and also have the capability to compile heavily mathematical algorithms of the sort yielded in chapters 3, 4 and 5. The hardware itself must have colour graphics capabilities enabling mapping of the geographical data to a high enough standard to allow spatial information to be conveyed effectively. Finally, thought must also be given to the operating system. The "control program" discussed earlier will be required to "freeze" and run other programs, and then restart on their termination; obviously the operating system under which the control program and its "child" programs are run must allow this to occur. Thus, choices must be given for the operating system, the machine to be used and the language for writing the software. Each of these will now be considered in greater detail.

6.2.3 Hardware Configuration

Firstly, the requirements of the application must be taken into account. As discussed above, coloured graphics and text are considered to be of

importance, in communicating spatial information and for more user friendly control menu systems. It is easier, when using multicoloured displays to draw attention to particular information, by highlighting it. Also disk space must be considered. Given the average space required to store data and programs, and the format in which they are stored, is it preferable to adopt a system with hard disk storage capability?

Another important factor is portability. If several different computer systems capable of running the same software are available, it is possible to develop software on one machine, but then run it on a different system. If the software is developed on a machine that is compatible with a wide range of alternative machines, then final choice of hardware is left with the user. Thus, according to budget constraints, and durability requirements and other factors (ie. some users may require portability of machine) the user may purchase one of a large range of suitable machines. It is also possible that, due to the ability of machines in a large family of "compatibles" to run a wide range of interchangeable software and to mutually exchange data files, such machines have already been purchased in a large organisation such as the police force.

Thus, considering the factors of video display capabilities and software compatibility, a reasonable choice for a micro to implement the software on would be either an IBM PC or compatible, on the condition that it has EGA (Enhanced Graphics Adapter) circuitry fitted. This machine will make a suitable candidate for a prototype system, since PC compatible machines are already widespread within Northumbria police force and

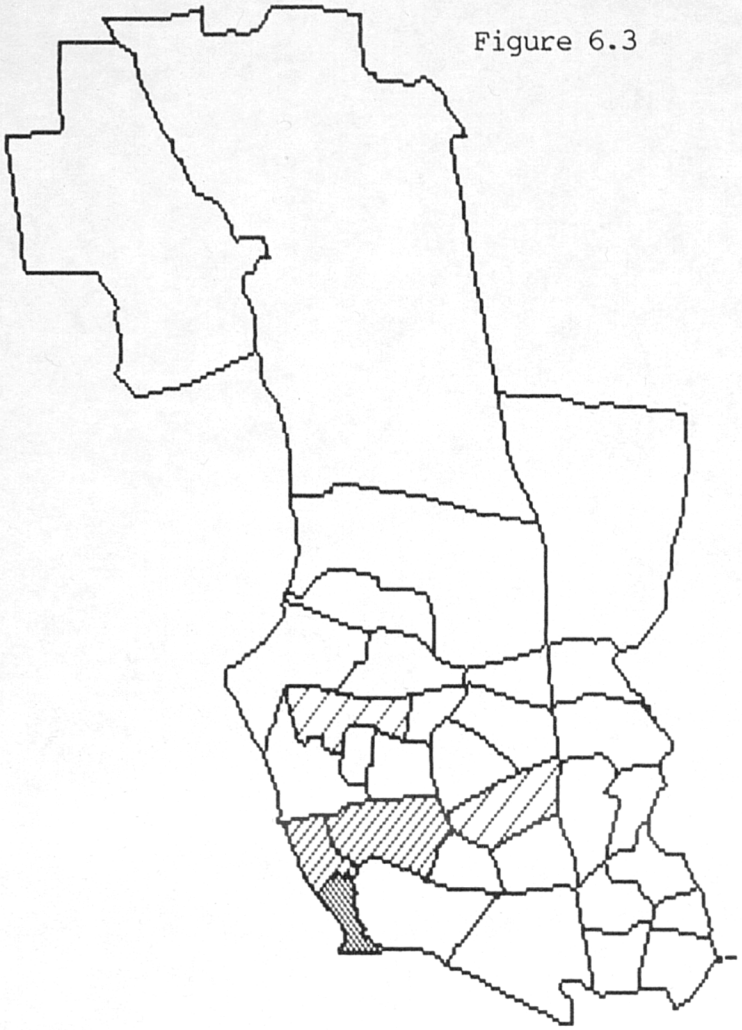
available as research tools within this university; thus software may be developed on machines in the research environment, and evaluated on machines in the work environment, without need of transporting any hardware other than the disks on which the program is stored.

Provided the system is fitted with an EGA facility, it is possible to obtain a screen having a resolution of 640 x 350 pixels, in up to 16 colours. The detail of mapping in this format is of reasonable standard, certainly capable of accurately displaying the information required here (see figures 6.3, 6.4 and 6.5). Again, many PC compatible machines are equipped with EGA graphics facilities.

A further justification for this type of machine is that, due to the wide range of software already available and adopted by a large range of users, there is some incentive to make future models of computer "downwardly compatible" so that they also run the software which will run on the current PC compatibles, although possibly faster due to hardware developments. Thus, any crime pattern analysis software developed in this hardware environment should run on future machines for some time. Hence, it is unlikely that software developed here will have to be drastically adapted to run on a different graphics hardware or a different operating system if, at a future point the prototype system is to be implemented as fully working.

Having decided on the type of machine, the configuration of disks, and memory must be selected. There are two types of disk drives generally available with PC compatible machines: 5.25" disks, 3.5" disks and hard

Figure 6.3



South Gosforth Subdivision
Household Burglaries
7 Days Ending 22/ 4/1984

0 -< 2
2 -< 4
4 -< 6
6 -< 8
8 -<
High Risk

Crimes per Week

Menu :-

<+> = advance 1 wk

<-> = go back 1 wk

<4> = 4 weeks data

<8> = 8 weeks data

</> =16 weeks data

<D> = High Risk

<F> = Exit to Menu

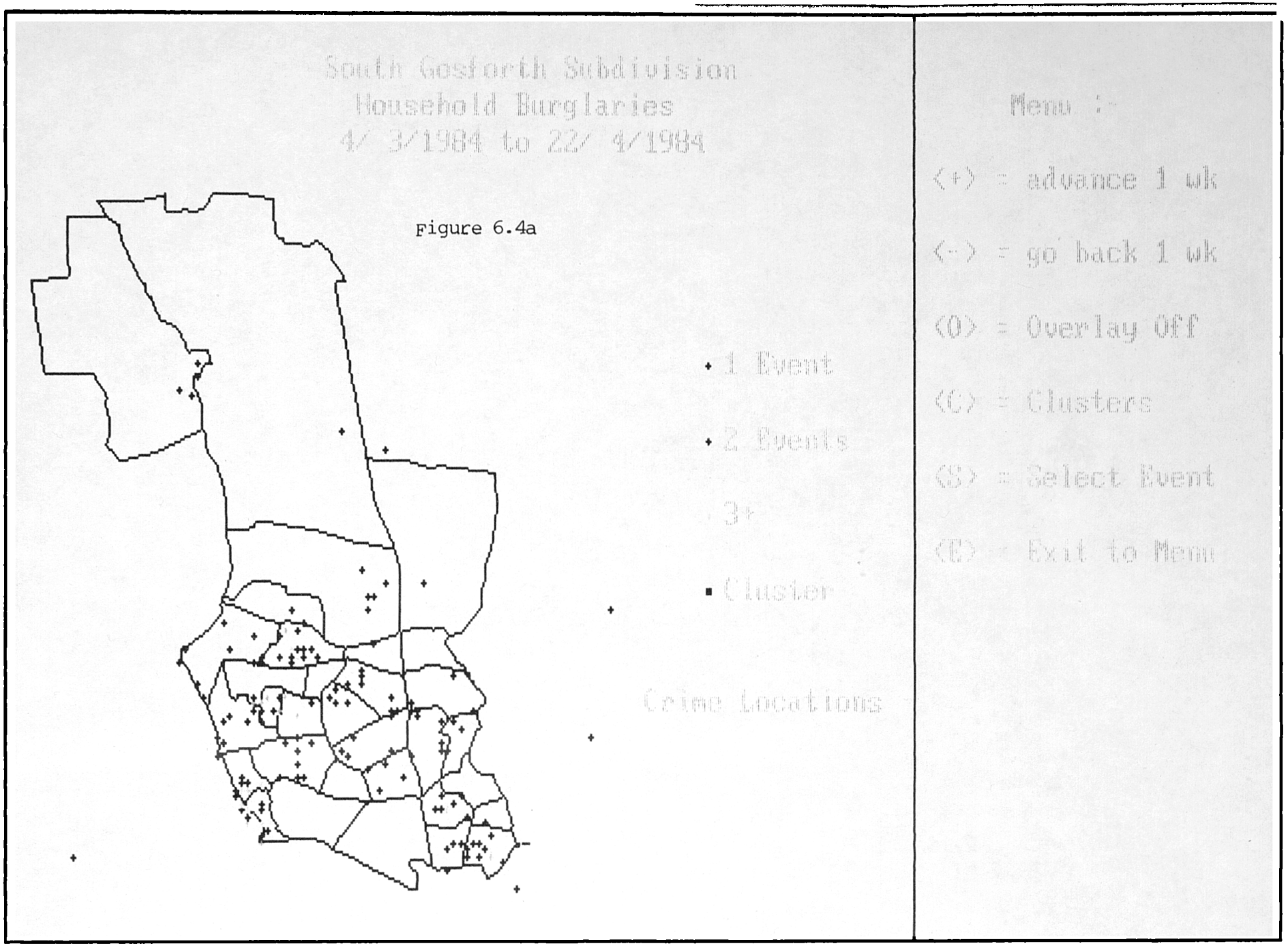

South Gosforth Subdivision
Household Burglaries
4/ 3/1984 to 22/ 4/1984
1 Event
2 Events
3+
Cluster
Crime Locations
Menu :-
<+> = advance 1 wk
<-> = go back 1 wk
<O> = Overlay Off
<C> = Clusters
<S> = Select Event
<E> = Exit to Menu

Figure 6.4a

South Gosforth Subdivision
Household Burglaries
7 Days Ending 8/ 4/1984

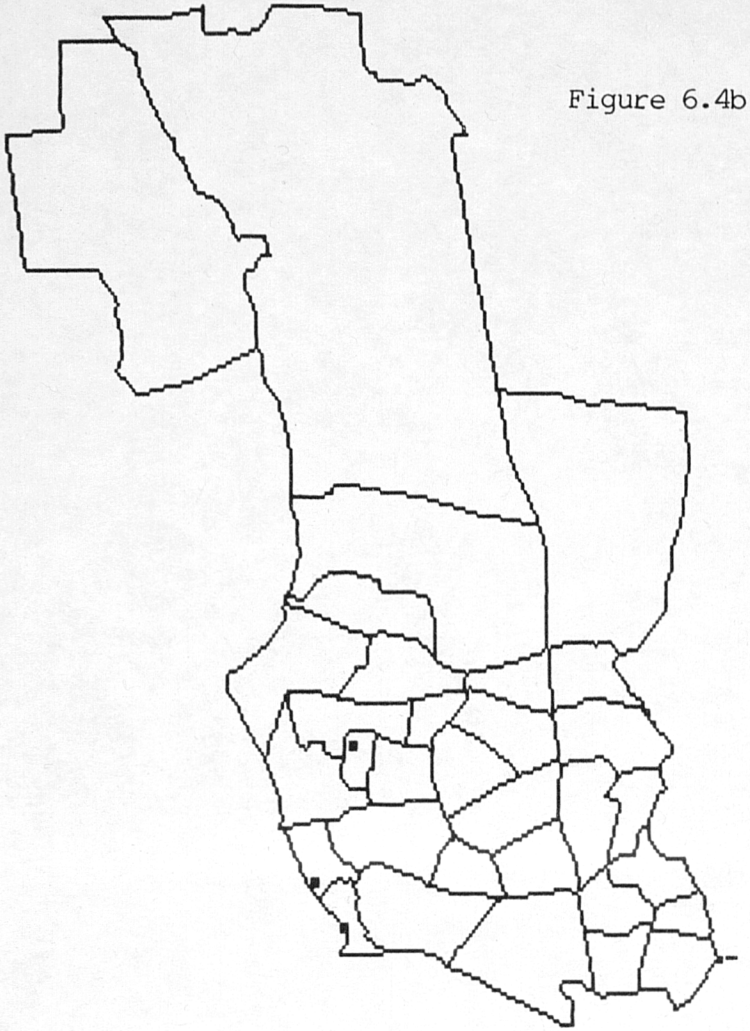


Figure 6.4b

• 1 Event

• 2 Events

3+

▪ Cluster

Crime Locations

Menu :-

<+> = advance 1 wk

<-> = go back 1 wk

<O> = Overlay On

<C> = Clusters

<S> = Select Event

<E> = Exit to Menu

South Gosforth Subdivision
Household Burglary Risk
16 Wks Ending 22/ 4/1984

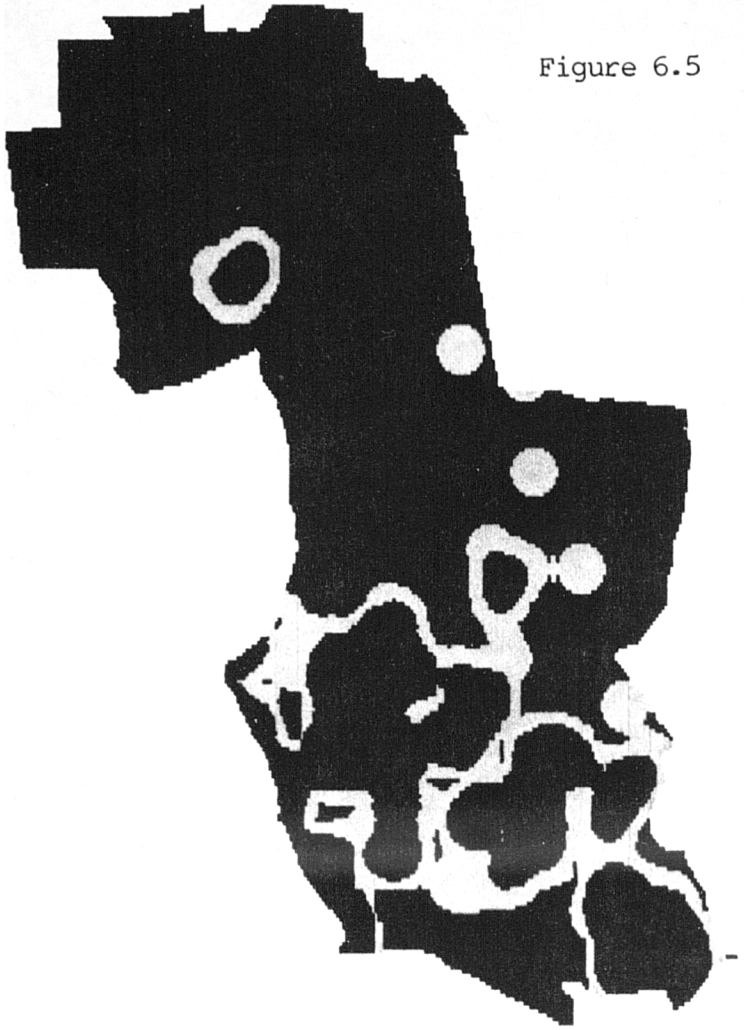


Figure 6.5

Low Risk

Med. Risk

High Risk

<B> = Beat Zones

<A> = Risk Zones

<E> = Exit to Menu

disk drives. Generally, their capacity in terms of amount of data in KBytes is listed in table 6.1. The hard disks, as well as containing much larger data reserves, are resident in the machine, and generally faster to access. However, these features considerably add to the cost of hardware. Although it is unlikely to be possible to implement the analysis system on a machine having only 5.25 inch disks (A single module often contains about 50K of code, and in addition to all modules some operating system code must be fitted on this software disk), it may be possible to do so on a 3.5" disk. It may also be possible to implement the system on two 5.25 inch disks, with one disk containing a mixture of data and code, but this would be awkward to implement - requiring code to be copied onto data disks - and deflating the "ease of use" objective.

It seems reasonable to keep the database on a removable disk. This allows greater security as the disks may be kept under lock and key when not in use, and also portability allowing the same data to be analysed at different sites if necessary. Thus, two main alternatives exist. Firstly, control software and modules on a hard disk, and data on a removable disk, or both on 3.5" removable disks. The 3.5" option is cheaper to implement, but it is possible that the combined size of all of the software modules may still exceed the capacity of the disk, or at least constrain the performance of the more data-intensive algorithms which may be slowed down by the relatively slow access time for floppy disks.

Table 6.1
Capacities Of Disk Storage Media

Disk Type	Capacity
5.25" Floppy	360 KBytes
3.5" Floppy	720 KBytes
Hard Disk	Between 10 and 70 MBytes

It is therefore intended to develop software on a hard-disk machine, but to bear in mind the option of a 3.5" disk implementation which will possibly have fewer options. Again, the modular design together with the menu descriptor file concept are useful here. By modifying the MDF on the 3.5 inch disk version, a restricted menu of options can be implemented without major alterations to any of the system software, apart from non-inclusion of some of the peripheral modules.

Another option offered with the PC compatible range is the inclusion of a hardware floating-point chip. This is an integrated circuit extending the instruction set of the CPU to include floating-point mathematical operations. When these are performed directly, as opposed to synthesised using bit manipulation techniques in machine code, speed of execution is increased considerably. Also, more compact code is produced, as fairly complex routines are replaced by single CPU operations. This is particularly important in a package such as this, employing several algorithms which are heavily reliant on floating point operations. Again, the cost of the extra hardware may rule out this option, but provision should be made for possibility of inclusion. This is a point that also should be considered when choosing a language to write the software. Different codes will be produced when compiling programs under the assumption of a floating point hardware facility, and this code will not run on machines without this facility.

Finally, peripheral hardware may be given consideration. As discussed earlier in the chapter, printout of tabular data may be required. Generally, if the output is required to contain only standard ASCII

characters, provided the hardware has a standard output port (ie. an RS232 compatible serial outlet) most commercially available printers may be driven. Also, although this would incur greater cost than the rest of the system, graphical devices such as thermal wax plotters may also be connected to such a standard port, to give a hard copy of screen dump. The sample maps (figures 6.3, 6.4 and 6.5) seen here are generated in this way, using a thermal wax plotter, with a supplied routine to copy EGA screens. Since this routine is of the "terminate but stay resident" (Duncan, 1986) nature, the crime prediction software may be temporarily "frozen" to allow EGA screen dumps, and then re-started.

6.2.4 Programming Language and Operating System

These two headings are considered together since in this application, interaction between the two is expected to be fundamental to the operation of the system. As has been discussed earlier, "freezing" of programs to transfer control to "child" programs will have to take place, as will interaction between the user and the menu and graphic displays. Generally, programming languages are designed to be system-independent, while operating systems are designed to control particular hardware configurations. Thus, unless the programming language is capable of directly accessing certain routines in the operating system, problems of implementation will ensue. It seems necessary, therefore, to consider both the requirements of the operating system and the ability of the programming language to access the particular system under the same heading.

Firstly, consider the operating system. Given the choice of hardware specified earlier in this section, one choice to be considered is MS-DOS.

This is command driven (rather than menu driven) and has facilities for "freezing" programs, as described above. The lack of a menu based front end is not likely to present problems in this context, as the crime prediction software is intended to provide this facility, with certain extra utilities specifically designed for this application, such as password protection for certain parts of the system.

MS-DOS may be accessed from within other programs by means of system interrupts, allowing direct access to input/output routines, and other system management code. In PC compatible machines, a similar set of interrupts allow access to graphics routines. Thus MS DOS is capable of interacting with the crime prediction software from within the routines. A further advantage of this system is that it is generally supplied as standard with most PC compatible machines (and recommended as a system for the PC itself) thus the crime prediction software can be installed directly onto the hardware system as bought, whereas, for example if a mouse or menu based operating system were to be used, this would have to be purchased on top of the basic system, before the prediction package could be run.

If it is decided, then, to write application software which will run on a system with MS DOS implemented, it is now important to choose a language to develop and write the software. As stated before, the language chosen should be capable of expressing fairly complex algebraic

algorithms in a reasonable format, and also have the ability to access the operating system. Further requirements also need to be considered. The schematic diagram for the forecasting system in chapter 4 suggests that certain algorithms used will be complicated in structure. Thus, a well structured programming language offering IF-THEN-ELSE, REPEAT-UNTIL and other similar constructs will be helpful. A language with these constructs should provide easily readable programs, which will allow translation of algorithms to code with a reduced error rate, and also enable faster trapping of erroneous code when it does occur. Programs of this sort are also more easily understood by other programmers, or by the author if they need to modify the code at some future point (Wirth, 1973 or Djikstra, Dahl and Hoare, 1972).

For its mathematical capabilities, FORTRAN appears to be a good choice, particularly as the standard includes an exponentiation operator, '**'. This is not provided in the standard definitions of Pascal or C. Although this is also offered in BASIC, this language can be rejected on other grounds, most versions are interpreted rather than compiled, so that execution speeds are poor. Also, named subroutines having arguments passed are not defined.

A major problem with FORTRAN 66 is its lack of program control structure. All decision based algorithms have to be specified in terms of "go to" statements. This has a tendency to make programs hard to follow, and certainly hard to modify. FORTRAN 77 goes some way to counter this: the IF-THEN-ELSE structure is included. Also FORTRAN 77 allows character manipulation and character operations in a more

natural format than the 1966 standard. In the latter, characters are stored in memory designated for some other type of variable, such as LOGICAL, and operated on by subroutine calls. Given that a certain amount of string processing occurs in the control program, and in the map drawing modules, it seems reasonable to demand that a more natural string processing method is implemented, again to reduce programming error, and to make error checking easier when problems do arise.

FORTRAN 77 has facilities to handle strings and algebraic expressions which exceed those of either Pascal or C. However, in terms of control structures, the latter languages offer better facilities, giving "DO-WHILE" "REPEAT UNTIL" and "CASE" structures. It is possible, however, to simulate both of these in FORTRAN 77 in a reasonably readable format (see table 6.2). If it is specified that, except under exceptional circumstances, the only use of GO TO statements will occur in these constructs, then readable, easily modifiable FORTRAN 77 programs should follow.

The final requirement of the language adopted for this project is that of relatively easy interaction with the operating system. Generally, this is a property of particular implementations of the language than of the standard definition. Clearly, the standard must be defined irrespective of the operating system, since such definitions are intended to be universal. However, some implementations of FORTRAN come equipped with library functions which interact with MS-DOS. One such mplementation is that supplied by PROSPERO. In this version of FORTRAN 77, the standard syntax is adhered to, but a library of

Table 6.2
Implementation Of Program Structures In FORTRAN 77

Structure (as In Pascal)		FORTRAN code

IF x THEN y;		IF (x) THEN
		  y
		END IF

IF x THEN y ELSE z;		IF (x) THEN
		  y
		ELSE
		  z
		END IF

WHILE x DO y;	100	IF (x) THEN
		  y
		  GO TO 100

REPEAT y UNTIL x;	100	y IF (NOT.x) GO TO 100

CASE w OF
w1: y1;
w2: y2;
  .
  .
  .
END CASE;
		IF (w .EQ. w1) THEN
		  y1
		END IF
		IF (w .EQ. w2) THEN
		  y2
		END IF
		  .
		  .

The last structure may also be represented in FORTRAN if w is an ordinal set of integers by GO TO (101, 102, ...), w with the label numbers referring to each case. Each statement should then be followed by GO TO 999, where 999 follws the last y-statement.

In addition, a 'menu' structure can be more efficiently implemented by a case structure combined with a repeat loop: the test at the top of the loop involves polling the user to make a menu selection, and often takes the form of several lines of code.

Note : y, y1, y2 and z refer to statements (in their FORTRAN or Pascal form), x a logical expression, and w is a variable of general type, with specific values w1 and w2.

subroutines interacting with MS-DOS is supplied. Descriptions of the most useful subroutines in this library are listed in table 6.3. Provision is made for operating system interrupts, reading text from the calling command line in MS-DOS and "freezing" while other programs run. With this library it is possible to interact with the graphics hardware, and build a menu-based control program as set out in section 6.2.

Thus, a system has been established, in which a microcomputer configuration with graphics facilities has been specified, together with an operating system, and a language to write the appropriate software. In addition to this, the mainframe computing facility at this university, an Amdahl 58/60 running the Michigan Terminal System (MTS) operating system, has a powerful interactive debugging facility for FORTRAN 77, so that, at least those parts of the software that do not rely on the operating system or the graphics interrupts can be developed on the mainframe using the debugging facilities, and then downloaded onto the micro. A stage has been reached, then, where the computer hardware and software development tools have been specified. It now follows to define the system itself.

6.3 System Specification

The design of the package will now be considered in more detail. Firstly, it must be decided exactly what facilities the system is to offer. The main aspects to be incorporated into the software are as below:-

1) Menu-Based Controlling Program

Table 6.3
Useful Library Routines In Prospero FORTRAN 77

GETCOM(character*(*))	Reads the MS-DOS command line and returns all of it excluding the program name, in a character variable.
EXEC(character*(*))	Causes the calling program to be frozen, the program whose name is in the character variable to be executed
SYSREG(array, int)	Causes interrupts to be generated. Interrupt number is in int, and array maps onto the registers of the CPU.

2) Displaying of Past Data

3) Prediction of future crime rates

4) Input of Incidence Data

5) Incorporation of "peripheral" software

Under heading 2), three sub-categories exist

2a) Choropleth Mapping of past data

2b) Point Mapping on past data

2c) Surface Mapping of past data

Also, under category 5) there are currently two sub-categories

5a) Time of Day of Burglary

5b) Seasonal Variations in burglary rates

However, results of any other "spin-off" research may be incorporated into this list of sub-headings. Clearly, heading 1) ties all of the other headings together, allowing the user to select from the remaining items on the list. Each of the headings will now be considered in turn.

6.3.1 The Menu-Based Controlling Program

This program provides the "front end" for the package. It is intended to make this as flexible as possible, so that new items may be added into the menu system or removed from it with relative ease. It is also important to make this software as robust as possible. In the case of an

error, control should not be given to the operating system, which the user may not be familiar with, but should return the user to the most recent menu screen. Also, the software should be robust to the user inputting incorrect menu responses. For example, if there are four items to choose from on the menu, corresponding to keys "1" to "4", the system should ignore any other key press, for example "+" or "k". In addition to these requirements security must also be considered. Certain parts of the system must be password protected. The intention to protect certain menus or programs could be conveyed in the menu descriptor files.

One record could consist of a single character, say "+" or "-", to decide whether password protection is required, and the remainder of the line used to store the password.

Some form of encryption should be used, otherwise it may be a relatively simple task for an unauthorised user to list the MDF and discover passwords. In this prototype, a relatively simple encryption method will be used - development of a powerful and secure encryption method is a research topic in itself - but in the future, a more complex method may be substituted.

6.3.2 Menu Descriptor File - Format

Consideration of the requirements of a menu descriptor file now leaves us in the position to define the format for such a file of data. This is shown in table 6.4. The "menu name" will be printed on top of the

Table 6.4
Format Of Menu Descriptor Files

Record Number	Description	Field Size
1	Title Of Menu	58
2	Title Of Menu Option	30
3	Help Line (Extra Information)	58
4	Password Line (If Preceded by + in field 1)	8
5	Action Preceded by a single letter M = Another menu E = Execute Program e = Execute and return to same menu S = Go to system	30

VDU, as a header. Then, for each item on the menu, a three-line descriptor follows. On the first line, the name of the program to be executed if the corresponding choice on the menu is made is given. Alternatively, the name of a new menu descriptor file may be given. This is chosen by the first character on the line. "M" implies that a new menu is to be referred to, "E" an executable code. A final option, "S", returns the user to the system. On the next line, the description of the program or menu is given. This will be the text printed on the VDU for the corresponding menu choice. The third line deals with password security. A minus sign indicates that no password is necessary. A plus sign indicates that a password is required: the encrypted password then follows. Thus, each menu item is stored in the menu descriptor file, and items may be added or removed using a text editor. The only problem now is the entry of encrypted passwords. Clearly, it would be useful to enter the non-encrypted version, and have the machine encrypt this automatically. In order to do this, one extra program, called "ENCRYPT" is written. This operates on the menu descriptor file, leaving all text alone except when it encounters a password record beginning with the symbol "*". This indicates a non-encrypted password follows. The menu descriptor file is then edited to give the previously specified format, ie. '+' followed by the encrypted text. This allows initial input of passwords in non-encrypted form, and encryption to follow this.

An example of a menu descriptor file is now given. In order to produce a menu of the form below:

Crime Pattern Analysis

1. Input Data Item
2. Examine Past Data
3. Predict Future Crime Rates
4. Exit to system

In which item 1 executes a program called "update", and items 2 and 3 display further menus, descriptor files called "Past. MDF" and "Pred.MDF" respectively, and in which no password protection is required, the menu description file given below should be used:

CRIME PATTERN ANALYSIS
UPDATE
Input Data Item

-

MPAST.MDF
Examine Past Data

-

MPRED.MDF
Predict Future Crime Rates

-

S

Exit to System

-

6.3.3 Operationalisation

The operationalisation of the menu system must now be considered. When the control program "boots up" the main menu should be displayed. Thus, some means of conveying this information to the control program is necessary. This may be done on the MS DOS command line. Any program in MS DOS is initiated by typing in its name as though it were a command: Thus if the control program were called "Menu", simply typing "menu" would initiate it. However, in the library for the particular version of FORTRAN 77 supplied, a routine for reading further text from the command line exists, called GETCOM.
"CALL GETCOM(X)" returns the remaining text on the command line into the character variable X. Thus, if the main menu descriptor file is called "MAIN.MDF", the command to initiate the menu program could be entered as "MENU MAIN.MDF" and "MAIN.MDF" could be transferred to a character variable and used as a filename within the program.

Another problem is how to "freeze" the control program and execute another program following the "E" on the MDF record. Again, in the supplied library, a routine EXEC performs this task: "CALL EXEC ("PROGX") would execute a program called "PROGX", and freeze the calling program, retain all of the current values of variables, and

positions of stacks. After "PROGX" had executed, execution of the calling program resumes. If "PROGX" returns an error, then instead of returning to MS DOS, control is returned to the calling program, together with a return code. An error message will be printed on the VDU. In this instance, should this happen, the control program will print a "press any key to continue" message, allowing the user to view the error message, before returning to the previous menu. Although the user may not be able to interpret this, it may be passed on to someone with a knowledge of the system.

Under some circumstances, it may be useful to return to the menu from which the program was selected (ie. when interacting between viewing past data and predicting ahead in a Bayesian framework), while under others, it may be more useful to return to the main 'root' menu. These could be controlled in the menu descriptor file, possibly by distinguishing between an upper or lower case 'e' in the first character of the action specifier. If, during the execution of a program, an error is encountered this should also be handled by the menu control program. It is possible that subtle bugs may occur in any item of software after a long period of time since the system is implemented. These should ideally not cause a crash to the main operating system, since this may cause problems to the user inexperienced in this aspect of computer message should be displayed, which may be of use to an expert consultant, but then, after waiting for a key to be pressed, the calling menu screen should return.

If the choice represents displaying a new menu, this is achieved by simply looping back to the menu display routine in the menu control program, re-loading the menu information from a new menu descriptor.

Finally, the option of returning to the operating system should be dealt with. This should normally be done only via a password option, to avoid accidental 'bombing out' into the operating system, and then the user being unable to restart the program, or worse, possibly destroying data files in an attempt to do so.

6.3.4 Security

In this section, some thought will be given to the implementation of a password system. As discussed above, it is important to encrypt passwords in the menu descriptor files. There are several ways by which encryption could be carried out. A relatively simple method will be implemented here, although beyond the prototyping stage more sophisticated methods will be adopted. Clearly, the final encryption technique could hardly be published here, if security of the data is of genuine concern!

Therefore, the method employed here will be relatively simple. FORTRAN 77 stores characters as 8-bit codes, which can also be interpreted as integers between -128 and 127. A pair of 8-bit codes may also represent an integer between -32768 and 32767. Some form of transformation of this larger integer onto another integer in the same

range would provide an encryption from one pair of characters to another. Clearly, this is considerably more powerful than a mapping of individual characters. The former would require only 26 mappings (36 if digits also incorporated) to be discovered, but using 16-bit integers, 26^2 (or 36^2) mappings must now be discovered. A requirement of this mapping is that it must either be reversible providing a unique code for each password or at least unique for any password character combination. Otherwise, encrypting both the true password and the guess may lead to correct matches even if the two are not the same.

In fact, it may be shown that a mapping which transforms the set of integers between -32768 and 32768 (or indeed any other set) onto itself, and does this uniquely (so that if the mappings of x and y are equivalent, then x and y are equivalent) must be reversible (see eg Birkhoff and Maclean, 1967).

Another important factor to consider here is how "recognisable" the encryption is from the original. For example, a transformation from merely swapping pairs of characters would meet the definition above, but hardly be of use. One possible approach may be to adopt a "bit scrambling" technique, in which a permutation of the 16 bits are returned after encryption. If there is a reasonable level of scrambling, encrypted data would be hard to decode. Alternatively, additive or multiplicative transformations of the 16 bit integers may be used (taking care not to cause overflow of variables, which may crash the program). The latter is more easily implemented in FORTRAN, working on a similar principle to a linear congruential random number generator (Hammersley

and Handscomb, 1964). The multiplier must be relatively prime to 2^{16}, to ensure the reversibility property (Knuth, 1968), but given this, a multiplicative encoding scheme may be used. This method is proposed here, because of its ease of implementation in FORTRAN. Obviously, integer multiplication is readily available, and FORTRAN also offers the option of addressing fixed memory cells as though they simultaneously contained data in both integer and character format.

Experimenting with various options lead to the conclusion that, at least for the prototype, a multiplier of 255 worked satisfactorily. This number is prime, and therefore relatively prime to any integer, and provides bit-shifting of a least six bits in the left-hand direction. Thus, this method will be employed. The code of the encryption program is given in listing 6.1.

Another aspect of security is making the control program crash-proof. Since access to the operating system allows the user to alter menu descriptor files, and possibly damage data files, it is important to protect against accidental (or deliberate!) crashes. These could occur for two reasons.

1) Errors in coding of the programs
2) User generated interrupts from the break key

In the case of programs run from the control program via "EXECPG" this is not a problem, as control is returned directly to the calling program. If there is an error in the calling program, this can be handled by a

routine QBREAK which causes control to resume elsewhere in the program, thus the user is not left in MS DOS command mode. Also, the routine SBREAK causes, after its execution, all keyboard interrupts to be ignored. If both of these facilities are used, and are called as the first tasks in the control program, then a reasonable degree of "crash proofing" will have been achieved.

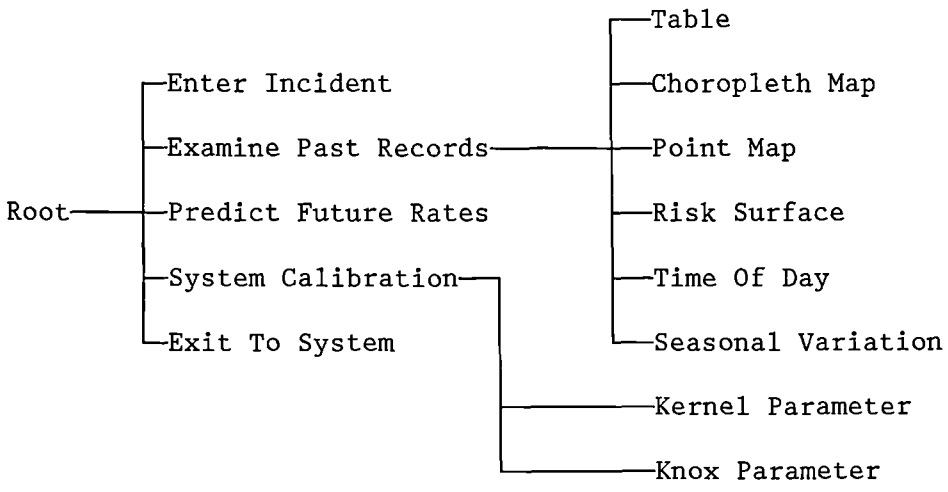
6.3.5 Control Program: Conclusions

The specification for a control program has now been given, together with methods of implementing various features considered to be necessary. The code for the main control program code is given in listing 6.2. The remaining issue with the control program is the actual structure of the menu system to be employed in the prototype system. This will be based on the recommendations in section 6.1. The basic structure will be a tree structure, with one option on each menu to return to its calling menu. This option is important, as unless it is implemented, the user is forced into executing some form of program eventually, and is unable to undo the consequences of a mistaken menu choice.

The tree is structured as in figure 6.6. The root node is the initial menu. The end nodes represent program execution, and the intermediate nodes represent transition from one menu to another.

Figure 6.6

Structure Of Menu system



6.4 Data Input Program:

This program is incorporated to allow reported household burglary data to be entered into the database. There are two main issues under this heading: the format of the data input screens and the format of data storage. The first is important, since if the user us unable to enter particular formats, or more likely to make errors inputting data, then the database upon which the pattern analysis depends is flawed, making any such analysis unreliable. The second issue is also of importance, as it is of consequence to all of the analysis programs that will be required to have access to the data. Both of these will be considered in detail.

6.4.1 Format of Data Input Screen

In a similar way to the menu based system, some flexibility must be introduced into the software for data input. Initially, the system will be used for the analysis of household burglary data. However, this may not be the only type of crime for which this sort of analysis is appropriate. Many of the techniques considered in this study may be applied to other spatially referenced crimes, such as car theft or vandalism for example. If this were applied to such crime types, it would be useful to be able to modify the data input program in some simple way, and allow this new type of data to be stored.

The format of the input screen could be similar to that of the menu screens, but taking the appearance of a form that has to be filled in. The "dotted lines" of a conventional form could be mimicked, with the user typing text over appropriate places on the screen. Care should be taken to ensure that it is not possible to type text outside of these

windows, since this would obscure part of the information on the form, and probably lead to confusion and errors from the person typing the data into the machine.

It is proposed to control the data input from an "input descriptor file" (IDF) in a similar way to the "Menu Descriptor Files" of the last section. These files give information of the data to be input on the screen, the text to be printed and the storage of this data. The format of the IDF will reflect the format of the input screen, consisting of the text on the input screen together with an "Image" of the data to be input.

It is also important to be able to correct data that has been incorrectly entered on the screen. Thus, provision to backspace on each "dotted line" must be provided, as well as a means of viewing the entire data screen, and verifying that this crime record as a whole may be committed to the database.

Before considering the specific format of the input screen, it is important to discuss the contents of the database, as these will govern the data items to be input onto the screen.

6.4.2 Storage of Data Relating to Crimes

In this section, it is proposed to specify exactly what data is required to be stored in the central database for this crime pattern analysis software, and in what format it is to be stored. Finally, the requirement of the analysis software are to be considered. This

software requires two main types of spatial data about crime: beatwise and pointwise. Pointwise data can be aggregated into beatwise data using point-in polygon techniques (Aldred, 1971). However, this is not a fast process, and it seems more practical to store data in both formats. If in a subdivision, the average crime count for household burglaries is approximately 40 per week, details of only 640 crimes will need to be stored over a 16 week period, and the file storage overheads of keep two separate files, one beatwise, the other pointwise, is virtually negligible.

It is also important to remember that, although the phrase pointwise is used here, the points are actually centroids of postcode units. The analytical effects of this have been addressed in Chapter 4, but on a level of data storage formats, is it best to store co-ordinates or postcodes. When a postcode is entered, it will be an 8-character code. Either this could be stored directly into the database, or converted via a look-up table into a grid reference. If conversion is not performed during storage, this must be done when the crime pattern analysis software is being run. It is felt that, in terms of ease of use, it is better to perform conversion at time of entry into the database. Some of the analytical software may already require large execution time, due to the floating point operations in heavily quantitative algorithms, and it seems more reasonable to spread the total delay time evenly over input and analysis, otherwise the operation of the analysis software will appear very lengthy.

Also, if conversion is performed at time of storage, it need only be performed once for each incident. Run time conversion would require each case to be converted from postcode to grid format every time an analysis was performed.

Thus, a need has been identified in the data input software to process the entries before storage on file, in terms of spatial referencing. Time referencing will be considered next. For predictive purposes, a weekly reference needs to be given to each crime for point pattern analysis, and for Knox testing (Chapter 4) each crime needs a daily reference needs to be given, since a certain amount of manipulation of date referencing is necessary, a character representation alone will not be satisfactory. Each event needs an integer reference for the date, allowing dates to be subtracted (to find days between events) or sorted into weekly classifications, or compaired using ".NE." or similar FORTRAN operators to base decisions on the chronological order of events. Such a mapping is provided by

$$d = 365\,year + date + 31(month-1) + Int((year-1)/4) - Int(\tfrac{3}{4}(Int((year-1)/100)+1)) \quad (month \leq 2)$$

$$d = 365\,year + date + 31(month-1) - Int(0.4\,month + 2.3) + Int(year/4) - Int(\tfrac{3}{4}(Int(year/100)+1)) \quad (month > 2)$$

which, given a year, month and date returns a single integer value (see, for example, Texas Instruments TI59 instruction book).

Thus, information for each event referring to time and space have been defined. This information will be sufficient for any of the methods given in chapters 3, 4 or 5, at least in terms of data referring to crimes. The only task remaining is to collate this data into two formats, one for

beatwise aggregated analysis packages, and one for grid referenced techniques.

Firstly consider the problem of storing count information for each beat on a weekly basis. If a look-up table is to be compiled for postcodes to grid references, this could also contain beat codes. Thus, when a postcode is entered, its beat code is also determined. This would also allow error trapping, since undefined postcodes, or those not in the subdivision, would be eliminated at this stage. The look-up table would need updating regularly with new postcodes. Eventually this could be done locally. Having found the beat, the week could also be determined, from the day number. This would be given by the nearest Saturday (or other day) before the date of the event. If the week at the oldest end of the database began on a day numbered x, and the day number of the Saturday closest before the event is y, then event is in week

$$\text{Int}\left[\frac{y-x}{7}\right] + 1$$

if the oldest week is numbered 16, and the most recent week is 1. Clearly, from the beat and week references, a count can be kept in a beat-by-beat crosstabulation file, which may be updated for each newly entered crime.

The point references may be stored on a similar week-by-week basis, with a list of x and y coordinates for each week. Along with this list of

coordinates, the day codes will also be provided to allow Knox-type analysis to take place.

This space-time referencing provides the analysis programs with sufficient information to run, but it may be helpful to provide extra verbal information for human users. This may then be referenced after computer analysis, to look for patterns in modus operandi and so on. If a crime reference number for each crime is also recorded in the lists of points and day codes, links will be possible between the crimes stored in the database, and any verbal information also referenced in this way. In the prototype, a single line record will be given to verbal description and stored in a third file.

This file may then be accessed by the pointwise program for analysing past data.

6.4.3 Data Input Revisited

It is now known which information about household burglary incidence is required for the database. The Input Descriptor Files can now be defined. Given a particular screen layout, there are four requirements for data input:

1) The postcode of the household
2) The date of the burglary
3) The reference number
4) Comments, etc.

Thus, the descriptor file could be made to look identical to the display screen, with some symbols to describe where each of these data items are to be input. The image provided could be plotted on screen with a boarder, to match the menu system, and then the data items interactively edited into the image. When the image is completed, the user can press the return key, to attempt to commit the record to file. If it fails, an error message will be printed. The types of error that may occur are:

1) Postcode error - either postcode incorrectly formed, or not in study area.
2) Numerical error - year, month or date of reference number contains a letter.

These will be reported, giving the user a renewed chance to enter the data item. An option to abandon is also required, to allow the user to exit the data input program if it was entered by mistake.

The symbols used to describe the input data in the image descriptor file may be combinations of letters or characters unlikely to be encountered in text on screen. Thus, the month section of the date, for example, could be symbolised by "$M". Likewise, the year and date parts could be symbolised by "$Y" and "$D". This leaves the postcode, "$P", the descriptive text "$T", and the crime reference number "$N". The default widths for each of these fields is set out in table 6.5. If any of the variables are not contained in the Input Descriptor File, the input program will be forced to exit, giving an error return code, while

Table 6.5
Default Widths In Data Input Screen

Symbol For Item In IDF	Description	Field Width
$D	Date	2
$M	Month	2
$Y	Year	2
$P	Postcode	8
$T	Description Text	60
$N	Reference No.	4

printing a message. This will cause the control program to apply its error handling code, rather than cause a subsequent crash, by some program attempting to access part of the database that the input program was unable to compile.

The program to manage the input screen, INPUT, is shown in listing 6.3. Note that the name of the Input descriptor file is communicated to the program using GETCOM, as is the case with the control programs access to the root menu file.

6.4.4 Data Input: Conclusions

A simple data input system has now been created. This is relatively flexible, so that, for example re-wording of the input screen is possible, if certain wording is difficult to understand, or if it is desired to alter the data base to one of spatial references to other types of crime. There are commercially available database management tools which may perform these tasks, but in order to reduce costs in the prototyping stage, the approach here will be used. Due to the modular design of the system, it may be possible to incorporate a commercially available data input program into the system, provided it can store data in a format easily and efficiently accessed by the prediction software. This may be convenient if the operators already have training in this package. Finally, as considered earlier, it may be that data collation is centralised to force headquarters,and that this program may in fact be replaced by software to poll data from central file serving equipment. However, the development of the small here program makes it possible to

operationalise the prediction an pattern analysis software at this research stage of the process.

6.5 Display and Analysis of Past Data

Under this heading, most of the mapping techniques incorporated into the pattern analysis system will be introduced. As set out in section 6.2, these will be required to display choropleth maps, point maps and contour-based maps relating to household burglary incidence. The data input related to crime incidence will be from the files as set out in the previous section: a list of points and dates, split by weeks for each reported occurrence, and a crosstabulation of counts on a week-by-week beat basis. However, in order for mapping to take place, further cartographic information is required. Some sort of file containing descriptions of the shape of the foot beat areas is necessary to allow mapping using the graphics facilities of the hardware. This first section discusses methods of encoding this information, and of plotting it onto the screen. Subsequent sections then deal with the specific problems of overlaying point, choropleth or contour information in conjunction with this.

6.5.1 Storage and Display of Cartographic Information

As seen in chapter 4, map display visualisation is helped by a certain amount of information of local geography being displayed. In the case of police beat officers, the indication of foot beat boundaries proved useful. Since these often ran along the paths of main roads, and individual

officers were familiar with the beats they had been patrolling, particularly in relation to road patterns, these proved valuable geographical reference objects.

In early map study, the GIMMS mapping package on a mainframe computer was used to produce various crime map formats, which police officers were asked to assess. However, this method of map production will not be available on the prototype system. Although a micro version of the package is available, maps produced in this way are not "interactive". For example, in the prediction software, the user is asked to modify prediction maps with subjective predictions; also it is desired to allow the user to interactively identify Knox clusters, and high risk beats. Given that the maps are so heavily interconnected with the analysis software, it is justifiable to write a set of map drawing subroutines to enter these features at points required in analysis. Also the transference of data, and entry/exit procedures required to frequently transfer control to and from the separate analysis and mapping packages may prove time-consuming.

This approach will provide a cost-effective and relatively fast means of map display.

Given this, some form of storing and plotting geographical data must now be devised. Their are two basic formats in which areal data may be stored. These are called vector and raster formats (Burrough, 1986). Briefly, in vector format, data to describe an area consists of a sequence of grid coordinates, which define its perimeter. A set of

sequences like this define a region, divided into areal units. In raster format, the region under study is divided into a grid, consisting of relatively very small cells. Each cell is assigned a value according to which region it is part of.

There are various advantages and disadvantages of both types of storage. The accuracy to which each area can be defined in a raster based system is dependent on the size of the grid squares. Thus, precision has a quadratic relationship with storage. In vector based systems however, increasing the frequency of points need only be linearly related to precision of definition. However, overlapping areas, or areas not fully covering the study area may arise due to errors in the specification of vector-based files. Also, point-in-polygon techniques, while being complicated geometrical algorithms in vector based systems (particularly if areas contain holes, or are not fully connected - eg. a system of islands) are simply two-dimensional look-up tables in raster based systems.

Generally, it is important to consider the input and output requirements of the mapping systems. Vector-based systems are very efficient at inputting digitised data, since this is virtually in vector format. Conversion using vector point-in-polygon algorithms must be carried out to obtain raster files. However, on raster based display devices (such as the EGA), raster storage is obviously more efficient at data display. Each line segment in a vector list must be converted to a raster line (usually by Bresenhan's algorithm, Bresenham 1965 or Wilton 1987) before display. Given that vector lists often contain several hundred

line segments, this could be time-consuming compared to the direct cell-by-cell copying to pixels offered by the raster based solution.

Given that, in this application, the maps will only need storing once (at least until beat boundaries are altered) but need to be displayed several times in a week, the rasterised storage option is proposed. This should be on a basis where each grid cell in the raster database corresponds exactly to a pixel on the VDU. Extra precision would be unnecessary, since it could not affect the display, and lower precision would give poor results: generally, "staircase" effects, where edges of grids appeared in detail, showing the inaccuracy of areal definition.

In mode 16 of EGA (Wilton, 1987) the entire screen is given by a grid of 350 x 640 points. A large "window" in this will be set aside for map plotting, the remaining screen used for interaction with the user, key display and other relevant information.

In its naive form, therefore, a raster storage format could be costly in terms of file space. However, a ""packing" scheme is now proposed which should overcome this problem.

6.5.2 Packing of Raster Files

Raster data specifying area units may be stored in an m by n array, where each element contains an integer which is used to indicate which zone that particular grid element is contained in. Generally, however, several adjacent grid cells will be in the same zone. Thus, the data in

a contiguous row could be replaced by two pieces of information: the zone code and another integer indicating how many times this code is repeated. If this format were generally adopted, a more space-efficient storage technique should follow: for good map definition, the size of the grid cells should be considerably less than that of the average zone size, so most cells would be expected to be in a contiguous row with respect to zonal classification. The computing overheads in doing this will be only a small increase on that of a direct grid-to-pixel mapping, only requiring the occasional initialisation of loops.

Also, since in these applications it may aid data organisation if the data for each foot beat zone were kept separate, a further modification may be made. Instead of treating the aggregation of all zones in a single file, each zone will have its own record. In this case, no zone number will be required in storage. In each record, the counts refer only to presence or absence of the zone related to the particular record. After an initial true/false specification, counts will be given for the number of continuous cells in a row in that state. The next number refers to the next contiguous count, of cells not in that state, and so on, until all relevant cells in that row have been considered, and the next row begins. The code for this could be, for example, -1, as this could not represent a count of cells. If the top right-hand corner cell is given, then an area could be defined as a list of the form above, being terminated by a pair of consecutive -1 values.

Thus, for each zone, the method continues the search through a window containing this zone. For each row a code is given to state whether the

leftmost cell is contained or not. Finally, when the rightmost cell within the zone to be defined has been given, a -1 code follows. This continues until the end of the zone, when a pair of -1 codes follow.

However, this algorithm, if given the data for an entire area, would plot a solid shape. There are times when only borders to areas will be required, for example when plotting point maps. Given the original n by m matrix, border detection is relatively easy. However, when using packed data, edge detection is more difficult, as there is no easy way of examining a cell's relationship to its upper or lower neighbours. This suggests that those pixels lying on areal boundaries should be identified when the packed file is being compiled from the full matrix, and be stored themselves in another packed file. The boundary of a zone may itself be thought of as a zone, of width one cell. Again, packed storage here will be compact, as boundaries will consist of either large consecutive runs of logical "trues" (on horizontal parts) or consecutive runs of "falses" (in the "hollow" parts of the boundary, ie. within the solid zone). Also, the same piece of software code may then be used to draw the solid areas and their boundaries.

The boundary cell detection rule, as put forward in the paragraph above, is relatively simple for matrix format data. Any cell contains an integer area code. If it is on a boundary, at least one of its four nearest neighbours will not contain the same area code as itself. Thus, it will be on the perimeter of the zone whose code it contains. In this case, however, double thickness boundaries would be plotted, since any pair of adjacent cells which were not in the same zone would both be

classified as perimeter cells. A better solution would be to only define as perimeter cells those cells whose index is greater than or equal to all of its neighbours. In this case, for each pair of adjacent boundary cells (according to the former definition), only one would be classified as a perimeter cell. This may give slightly inaccurate results when several layers of areas one cell thick arise (if the central layer codes do not exceed those sandwiching) but in terms of visual display, these effects will only occur at a resolution of one Pixel, which should be small compared to the overall scale of the map.

For each zone, perimeter cells are stored in packed format. Note that, if plotting an individual zone perimeter, the entire zone may not be enclosed (due to the "greater than or equal" formulation above) but if all zones are plotted from this file, all boundaries will be displayed. The zone-by-zone format merely ensures compatibility with the solid zone packed files.

Some thought must now be given to the conversion of vector files, from digitised output, to packed raster files. This is best done in two stages: firstly from digitised output into the raster matrix, and secondly from raster matrix into packed raster format. The first is relatively simple. For each grid cell, an assignment rule to an area must be specified. Here, the grid cell centroid will be tested to see which polygon it lies in, using point in polygon techniques. If it lies in none, it is assigned zero, otherwise an integer code.

At this stage, a second matrix is created, and its elements represent perimeter cells for each area, as outlined previously. Both of these matrices may then be packed, by scanning row by row for continuous runs of cells, and outputting the run length data as specified above, for each area code in turn. The code to carry out this procedure is given in listing 6.4.

Initially the file containing packed data will be in text format. This is because at first the conversion from vector to packed raster data takes place on the mainframe computer installed at the research site. This text file may be downloaded to a micro, and converted into binary data. As binary files may not take the same format on both mainframe and micro, it is important to postpone text-to-binary conversion until after transfer. The initial conversion takes place on the mainframe for two reasons. Firstly, in the intermediate raster matrix stage, storage overheads may be higher than practical for the current micro, also the perimeter cell detection may be expensive in CPU time for a micro. Secondly, the vector data exist on GIMMS dump files (Waugh, 1981) which are resident on the mainframe, and thus more easily accessed.

Listing 6.5 gives the code to convert the output of this conversion program into a binary file; and listing 6.6 gives a subroutine to plot a file containing a collection of areas onto the screen. This calls two other subroutines, MODE which initialises the screen in a given graphics mode, and DOT (I, J, K) which sets a Pixel with coordinates I and J to colour K (NB colour 15 is white). Finally, figure 6.7 gives a thermal wax screen copy of the display given by the subroutine.

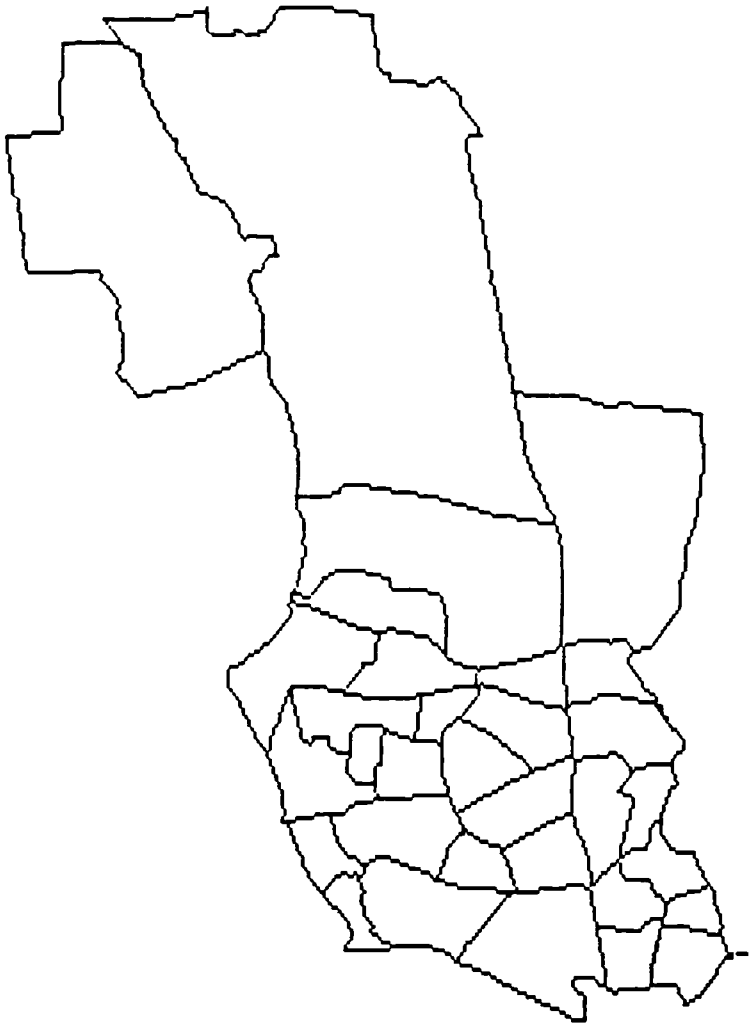


Figure 6.7

6.5.3 Choropleth Mapping

Having devised a method of mapping beat areas or their boundaries, methods of combining these with the data must now be considered. Firstly, choropleth mapping techniques will be considered. In this technique, an aggregate value for an entire zone is represented by the shading pattern for that zone. Already an algorithm exists (algorithm 6.1) to shade a solid area uniformly in one colour. This could already *be used to shade areas in terms of past* crime rates, simply by filling solid areas out in different colours. In EGA mode 16, there are 16 colours available. However there is no obvious ordinality in the set of colours offered, so using this technique directly would not produce an intuitively informative map. It would be better to classify all of the beats by shading in the same colour, but distinguishing different rates by using hatching of different pitch. There is easily recognisable *ordinality in this*, if pitch is ordered in the same way as crime rate. This is done here by modifying the method set out in algorithm 6.1; the "dot" routine is called selectively according to a condition on the sum of the x and y pixel coordinates. This condition is that

$$x + y \nmid 0 \mod K,$$

ie. $x + y$ is a multiple of k. The higher the value of K, the greater the pitch of the hatching is. Since this rule directly relates to pixel positions as integers, this does not result in the jagged edges and rounding errors characteristic of hatching based on real numbered pitches. It also provides a faster algorithm than vector based hatching

algorithms, only adding a single filter to the block fill routine to decide whether pixels should be plotted.

The full plotting subroutine is given in listing 6.7. This allows colour and pitch to be adjusted for each beat. The change of colour will be required to highlight beats having surprisingly high rates, given the values of their neighbours.

Thus, the basic tools for the beatwise map of past data have been created. These building blocks may now be joined to provide an interactive program. There are options to view choropleth maps for single week periods over the past 16 weeks, from the current week. Also options to view the data aggregated over 4, 8 and 16 weeks are offered. Finally, records will also be kept for beats which had "surprisingly" high beat rates (see chapter 5 section 4). An option will exist to view these by highlighting in a different colour. Initially pitch shading will be light blue, with "surprising" beats marked in red. A listing of the complete choropleth mapping program for past data is given in listing 6.8.

6.5.4 Point Mapping

The aim of point mapping of past data is to overlay point markers showing the locations of household burglaries onto beat boundary outlines. This is to be done on a week-by-week basis. It may also be useful to overlay the points for several weeks, to build up point

patterns over a longer period of time. Finally, it will also be of use to identify clusters that are close in both space and time.

Point plotting is relatively simple. Assuming data is read from a file of past point estimates as described in the input program section, the coordinates are linearly transformed to pixel coordinates. These are then plotted on the VDU. Early experimentation of point plotting programs drew attention to two difficulties. Illumination of single pixels did not prove to be a good means of displaying the data, as they were difficult to discern, particularly near to borders. It was therefore decided to mark points with crosses, built out of five pixels, as shown in figure 6.8. The second problem encountered was that of overprinting. Given the resolution offered by the VDU, and the fact that houses are postcode referenced, with several houses per post-code, several houses per postcode, several household burglaries may be allocated to the same point on the screen. This gives the appearance that only one incident has occurred, when in fact several have done so. The result of this could be that certain crime patterns may be obscured. To counter this, when a point is initially plotted, it is done so in red. A second overlay is in magenta, and three or more in yellow. Thus the use of the colour display may be used here to compensate for shortcomings of resolution.

Next, a means of identifying Knox clusters on the display is proposed. The intention is to highlight burglaries that have occurred within a certain distance of their nearest neighbour, and also within a certain time of their most recent temporal neighbour. Choice of distance and

time limits were considered in chapter 3. Reasonable limits are within 200m and 1 day. When identification of knox clusters is requested, for the week on display it is intended for each crime record to search all other records on the day associated with that record, and the previous day, for other events within 200m. On the first day of the week, the previous days records are to be found in the database for the previous week. Also, for the final day of the week, (if it is not the most recent week), a search will be carried out on the first day of the next week. Burglaries of this sort may be part of a cluster overlapping weekly *boundaries, and it would be dangerous to systematically* exclude them.

Thus, events that are components of this type of cluster will be identified by the algorithm. These will in turn be plotted on the VDU, in another colour from the colour coding discussed above, over the point incidence data. To further emphasise these points they will not be plotted with the shape given in figure 6.8, but the larger shape of figure 6.9.

Finally comes the problem of overlaying points, to gain several weeks of data on the same map. The approach taken here is to overlay these again using colour codeing for duplicate pixels. When the option "overlay" is selected, stepping back in time on the graph causes the past data to be plotted on top of current data, rather than having current data erased. If the mode is switched off, data is erased when the user steps back and forth in time, as before. The listing of the pointwise program is given as listing 6.9, and a thermal wax copy of the screen is given in figure 6.4.

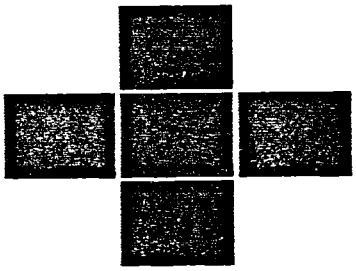


Figure 6.8

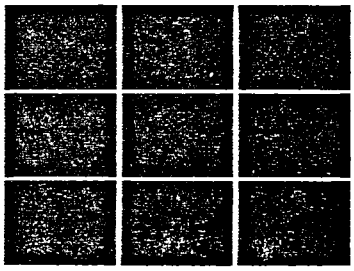


Figure 6.9

6.5.5 High Risk Area Identification

In Chapter 4, it was seen that, using kernel estimation techniques, a "risk surface" could be built up over a geographical study area, given a set of coordinates for the incidence of spatially referenced crimes. The points at which this surface exceeded certain values could then be thought of as high risk areas. Clearly, this would be a further useful method of representing past data. Indeed, in the map perception survey in chapter 7, a notable proportion of those surveyed rated this form of display highly, some commenting that it identified areas of high crime incidence that crossed beat boundaries.

A method of this sort could be implemented on the prototype system, when some constraints have been considered. The full Kernel estimator system, as implemented in chapter 4, would be difficult to code in a way that did not require very large run times: this is due to the large computing overheads required for floating point operations, several of which would have to be computed for each crime incidence point. Basically, if an incident occurred at a point x_i, then a kernel distribution function of the form

$$\frac{1}{nh}\sum_{i} g\left(\frac{x_i - x}{h}\right)$$

would need to be computed at several points on a lattice surrounding x. The process could be speeded up, however, by noting that this function is the same for all x, only depending on the distance between x, and a point on the lattice k. In this case, the function need only be computed

once, for various radius values. If each location of crime is rounded to the nearest pixel integer coordinates, and each grid point at which the surface value is to be computed also corresponds to a pixel, there is only a small number of distances for which the value of g need be computed. They correspond to the kernel function evaluated on a regularly spaced grid at which the reported crime is the origin. By symmetry, only the first quadrant of the grid need be considered. Also, beyond a certain radius, the value becomes negligible (NB, the spacing of the grid corresponds to 32m in the x direction and 40 in the y). Thus, at some point, this matrix of values can be stored on file, and referred to during the kernel estimation program. Execution may be further speeded up by storing these values as integers, say between 0 and 100; this allows the adding up of contributions at grid squares from several crime points to be speeded up, by performing no floating point operations.

Eventually each cell will have a risk "score". The file containing the kernel estimation matrix will be created externally to the running of the day-to-day system. A program to do this, in listing 6.10, is supplied. This allows several different shapes of kernel estimator, with choice of bandwidth (see Chapter 4) to be generated. This program may not be directly accessed from the menu, but could be run separately from MS-DOS, as part of an initiation process when setting up the system. *Alternatively*, it could be accessible from a password protected section of the menu, so that accidentally changing the values of the Kernel array elements is unlikely.

The graphical display of risk surfaces is relatively easy, when areal representation is raster-based. In a matrix, the risk scores for each pixel (corresponding to a rectangle of ground of dimension 32 x 40m) are stored. Pixels below a certain level will be plotted as green, and those exceeding it will be plotted as red. Thus, high risk areas will be indicated as red. In a similar manner to the hatching, the DOT routine is modified to colour pixels according to this score. In fact, a three-scale shading is employed, with yellow representing a medium score.

As in chapter 4, there is no simple way to decide the crossover scores from medium to high, or low to medium. Initial values have been decided on a trial and error basis, where high risk zones have been calibrated to cover areas which, on the opinion of some consulted police officers, are of a reasonable size to allow police manpower resources to respond. With the methodology set out as before, it is now possible to write code to perform the above tasks. This is given in listing 6.11. In addition to those requirements above, it was also decided to allow beat boundaries to be drawn either obscured by or superimposed on the risk area shading, which allows the user to identify areas of overspill between beats, by firstly considering maps without beat boundaries, and then overlaying these boundaries.

6.6 Prediction Software

So far, the software concerned with the display and analysis of past crime data has been considered. However, an important part of the

system is concerned with beatwise prediction of crime rates in the short term. In this section, the quantitative algorithms concerned with Bayesian forecasting will not be considered, as the operationalisation of the Bayesian system was considered in chapter 5. It will be assumed here that subroutines exist to perform this, directly translated from the algebraic expressions arrived at in that chapter. The main task of this program is to provide an interface between the human user and the Bayesian system, capable of translating between the probability based requirements of the system with the users beliefs relating to crime rates and their knowledge of policing the area, which are intrinsically expressible as formal probabilities.

In addition to this, the program will be required to output maps of predictions made, based on the human user predictions combined with the computer based predictions. Since these maps are also based on probability distributions, some conveying of uncertainty must also be incorporated into the maps. Beats may have point predictions based on mean values for the relevant variables in the posterior distribution (see Chapter 4) but it may also be informative to display the variances in some form. This should not be a direct map of variance for each beat, since this may prove difficult to interpret for some officers (Chapter 2) but could, for example, highlight those beats whose prediction distribution variance exceeds some limit as "less predictable beats". Also, lines can be drawn between beat centroids (stored in a "local information file") when beats are strongly correlated in the predictive distribution.

Firstly, the input of subjective knowledge from police officers will be considered. It was stated in chapter 5 that it may be helpful to show officers an initial prediction, and allow them to adjust this. This is reasonable as it allows adjustments of mean values of the predictive distributions to be made. However, it may also be useful to discover the degree of certainty that the officers place on their predictions. Again, it would not be reasonable to ask this in a statistical manner, such as requesting the input of standard deviations; therefore a three-way multiple choice question is asked; "How surprised would you be if next week's rate differed from this?" with options "not very", medium", "very". These can then be translated into variance figures. These figures can then be converted into normal distributions by using the number input as the mean and then choosing the standard deviation to reflect the answer to the second question. In the prototype, the following values will be used: for the response "not very" the standard deviation will be two times the mean, for "medium" equal to it and for "very", half of it. These details will then be fed to the prediction subroutine.

As suggested earlier, high variance in the prediction distributions could be indicated in a similar way to "surprising" beats on the past data choropleth program. The predictions could be shown as a single colour hatched choropleth map, and if it is requested to show "unpredictable" beats, these could be highlighted in another colour. Beats whose predictions are correlated could also be shown, by joining their centroids with lines. These lines, unlike hatching, would require Bresenham's

algorithm (Bresenham, 1965) to draw them. Thus, in addition to the straightforward choropleth map, there will be three options:

Firstly, an option to enter subjective beliefs. Secondly, an option to highlight beats with high variance in their predictive distributions, and thirdly to highlight strongly associated beats (ie. those beats whose predicted crime rates are strongly correlated in the Bayesian forecasting distribution).

The prediction program is shown in listing 6.12. It was also decided to incorporate a list form of beatwise forecasts, on the user modification option. This may be screen dumped, to get hard copy text output to circulate among police officers who may be concerned. A companion program to this updates the performance evaluation function of the human forecaster (see Chapter 5). This is called when a new weeks data is loaded onto the database by the Data Input Program. This is given in listing 6.13.

6.7 Miscellaneous Other Options

In addition to the main data input, mapping and prediction software, some other software is also to be incorporated into the package. As discussed earlier, this will not be considered as "mainstream" to the application, and will initially require data from files that are not dynamically updated using the data input program. However, they may be useful on an on-site evaluation by police officers to identify further development directions. Two such options are included. The first

relates to the "time of day" study in Chapter 3. The method reads in data in the format specified in that chapter, and from this deduces those times of day at which household burglaries are most likely to occur. The program specified called DAYTM, carries out this analysis, and data is coded by day also, and the program offers the further option of working out risk profiles for single days of the week, and also for week ends and week days separately.

The second option allows the total number of household burglaries in the subdivision to be analysed as a weekly time series. It is not the intention here to facilitate Box Jenkins analysis, or other complex techniques to be applied to the data. It is simply to provide seasonal pattern analysis. In previous studies this has often been found useful for medium term planning (ie. with a horizon of say three or four months). Two options will be offered here: a simple bar graph of weekly crime rates over the past year, and also a cumulative graph. In the cumulative graph, the option of comparing this year with the previous year is incorporated, to allow the current years performance in crime prevention to be assessed. At the end of each month, a "target" could be given to keep the years cumulative total no worse than that of the previous year.

6.8 File Structure

Now that all of the software in the system has been designed, and the structure of this software has been given in terms of the menu system, it may now be useful to consider the interaction of this software with the

data files. These are often the means of communication between separate programs, and it is important to consider firstly what files will be necessary and secondly which software writes to and reads from the files. The structure is best illustrated in table 6.6.

There are two stages. In the initiation stage, most of the data files are set up. These are listed in table 6.7. They include data about beat boundaries, name of subdivision continuity of beats, beat centroids and inter-centroid distances. It is expected that once they have been set up, they will not need to be altered in day to day running. Occasionally one may need to be altered by someone with access to the system, for example if a beat boundary is altered.

Secondly, there is the structure of day to day running. In this set up, the files set up in the initiation are not altered; they are only used for reading data. There are, however several dynamic files. These include point and tabular crime rate data, and data referring to the Bayesian prior distributions, which are modified as further data evidence is gathered. Generally, the data input program writes to these files, but other programs read them. The exception to this is the prediction program, where users prior predictions are output, so that there is a running record of their performance.

Table 6.6
File Interdependance

Program Name	File Number																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Menu Controller	*																			
Input Crime Incident		*	*	*	*	*														
Point Data Mapping				*		*	*	*	*											
Risk Surface Mapping				*				*	*	*										
Choropleth Mapping			*					*	*		*									
Prediction			*					*	*			*	*	*	*	*	*	*	*	*
Kernel Function Setting										*										
User/Machine Corrector			*								*			*						

File Numbers

1	MDF	Menu Descriptor File
2	IDF	Input Descriptor File
3	TABCRM	Beatwise Crime Dataset
4	SPTCRM	Pointwise Crime Dataset
5	POSTCODE.SQZ	Postcode to Foot Beat Lookup
6	TXTCRM	Text crime description
7	KNOX.BIN	Definition of Knox Clusters
8	BEATS	Beat zone descriptions for map drawing
9	BORDERS	Beat Border descriptions for map drawing
10	KERNEL.BIN	Kernel Function For Risk Surface Evaluation
11	COMP.MON	Monitor of Computer Predictions
12	DISTS	Distances Between Beat Centroids
13	ADJLST	Adjacency List
14	HHOLDS	Beatwise Household Counts
15	STAR	Space Time Autoregression Coefficients
16	BTMEAN	Beat Mean Levels Of Crime Estimates
17	COMP.PRD	Computer Predictions
18	USER.PRD	User Predictions
19	USER.PER	Performance Of User Predictions
20	CENTS	Centroids Of Beats

Table 6.7

Files Requiring Initiatiation Before System is Installed

1	MDF	Menu Descriptor File
2	IDF	Input Descriptor File
3	KNOX.BIN	Definition of Knox Clusters
8	BEATS	Beat zone descriptions for map drawing
9	BORDERS	Beat Border descriptions for map drawing
10	KERNEL.BIN	Kernel Function For Risk Surface Evaluation
12	DISTS	Distances Between Beat Centroids
13	ADJLST	Adjacency List
14	HHOLDS	Beatwise Household Counts
15	STAR	Space Time Autoregression Coefficients
20	CENTS	Centroids Of Beats

6.9 Conclusions

In this chapter, a set of requirements for a microcomputer crime pattern analysis system has been drawn up. Following this, an implementation has been proposed, and software to perform the specified tasks has been written. This software has been tested for errors by the author at the research site. However, it is possible that further flaws may become evident when this prototype is tested in a working situation. it is also possible that certain aspects of design, although perhaps implemented with the intention of being easy to use and relevant, may in practise prove not be so. At this stage, it is therefore necessary to set up an end-user based trial, with the software to be evaluated by members of the police force. In this way, the intended users are given an opportunity to change certain aspects of design in the development stage. In the following chapter, the methodology, implementation and results of such a trial will be discussed.

LISTINGS FOR CHAPTER 6

```
C
C       *****  Listing 6.1 ********
C
C   Encrypt passwords in an MDF
C
C
      CHARACTER*30 CHOICE(8), PASSWD(8), TNAME(8), NMDF
      CHARACTER*1  PWTYPE(8), TTYPE(8)
      CHARACTER*58 EXPLAN(8), HEADER
C
C Attach menu descriptor file
C
      CALL GETCOM(NMDF)
      OPEN(1,FILE=NMDF)
C
C Read its contents
C
      I = 1
      READ (1,1) HEADER
    1 FORMAT(1X,A58)
    4 READ (1,5,END=3) CHOICE(I)
    5 FORMAT(1X,A30)
      READ (1,1,END=6)        EXPLAN(I)
      READ (1,2,END=6)        PWTYPE(I), PASSWD(I)
    2 FORMAT (A1,A30)
      READ (1,2,END=6)        TTYPE(I),  TNAME(I)
      I = I + 1
      GO TO 4
    3 ITEMS = I - 1
      CLOSE(1)
C
C  Now encode the passwords
C
      DO 100 I = 1, ITEMS
  100   CALL ENCRPT(PASSWD(I))
C
C  And output the result
C
      OPEN (1,FILE=NMDF)
      WRITE (1,1) HEADER
      DO 200 I = 1, ITEMS
         WRITE (1,5) CHOICE(I)
         WRITE (1,1)       EXPLAN(I)
         WRITE (1,2)       PWTYPE(I), PASSWD(I)
         WRITE (1,2)       TTYPE(I),  TNAME(I)
  200 CONTINUE
      STOP
C
C  Error trap for badly formed menu descriptor file ...
C
    6 WRITE (6,*) 'Unexpected end on menu descriptor file ',NMDF
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE ENCRPT(PW)
C
C Password encryptor.  Currently crude, for prototype
C
```

```
      CHARACTER*30 PW
      INTEGER*2    PWINT(15)
      EQUIVALENCE(PWINT, PW)
      DO 100 I = 1, 15
C
C  Get 4 byte representation of the 2 byte section
C
        J = PWINT(I)
C
C  Encrypt it
C
  100   PWINT(I) = MOD(J*255, 32768)
      RETURN
      END
```

```
C
C          ********* Listing 6.2 *********
C
C
C  Control program for police software.  This program displays menus,
C  and then,  having recieved a choice from the menu,  it either
C  runs a selected piece of software or displays a new menu.
C
C
C
C        INPUTS  --  Menu Descriptor File on Channel 1
C                --  Keyboard (Via MS/DOS)
C
C
C        OUTPUT  --  VDU (Via MS/DOS)
C
C
C        EXTERN  --  Loads 'Child' Programs (Via MS/DOS)
C                    (Names specified in menu file)
C
C  ***  MS/DOS compatible only - Also some 'Child' programs require
C  ***  EGA graphics board in IBM/PC compatible machines.
C
C  Subroutines Called -- GETKEY  waits for a keystroke and returns it
C                     -- MENU    draws menu on screen
C                     -- CHOICE  makes menu selection
C                     -- EXECPG  Dos 'Child' initiator
C                     -- GETCOM  gets initiating command line from Dos
C
C
      CHARACTER*30 CHOICE(8), PASSWD(8), TNAME(8), IMENU, CMENU, PWDEC
      CHARACTER*1  PWTYPE(8), TTYPE(8)
      CHARACTER*58 EXPLAN(8), HEADER
      INTEGER*4    ITEMS, ICHCE, ERCODE
      LOGICAL OK
      INCLUDE 'A:SYSREG.FOR'
      COMMON /MVAR/ CHOICE, PASSWD, TNAME, PWTYPE,
     1              TTYPE, EXPLAN, HEADER, ITEMS
C
C  Disable break for security
C
      CALL SBREAK
C
C  Find the root menu
C
      CALL GETCOM(IMENU)
      CMENU = IMENU
C
C  Display it --- Main Menu Loop
C
  100 CALL   MENU(CMENU)
C
C  Make selection
C
      CALL CHOOSE(ITEMS,ICHCE)
C
C  Help line required
C
      IF (ICHCE .EQ. 0) THEN
        CALL HELP
```

```
        GO TO 100
      END IF
C
C  Go to new menu
C
      IF (TTYPE(ICHCE) .EQ. 'M') THEN
        CALL SECURE(ICHCE, OK)
        IF (OK) CMENU = TNAME(ICHCE)
        GO TO 100
      END IF
C
C  Execute a program
C
      IF (TTYPE(ICHCE) .EQ. 'E') THEN
        CALL SECURE(ICHCE,OK)
        IF (OK) CALL EXECPG(TNAME(ICHCE), ERCODE)
C
C  Check it ran ok
C
        IF (ERCODE .NE. 0) THEN
          WRITE(6,*) 'Press any key to continue ... '
          AH = $08
          CALL SYS1(SYSREG)
        END IF
        CMENU = IMENU
        GO TO 100
      END IF
C
C  Run a program but do not go back to root menu
C
      IF (TTYPE(ICHCE) .EQ. 'e') THEN
        CALL SECURE(ICHCE,OK)
        IF (OK) CALL EXECPG(TNAME(ICHCE), ERCODE)
C
C  Error check
C
        IF (ERCODE .NE. 0) THEN
          WRITE(6,*) 'Press any key to continue ... '
          AH = $08
          CALL SYS1(SYSREG)
        END IF
        GO TO 100
C
C  Exit to system
C
      END IF
      IF (TTYPE(ICHCE) .EQ. 'S') THEN
        CALL SECURE(ICHCE, OK)
        IF (OK) THEN
C
C  Reset break interrupt to normal and stop program
C
          CALL RBREAK
          STOP
        END IF
        GO TO 100
      END IF
C
C  Error in MDF
C
      WRITE (6,*) 'Non standard action specifier in ',CMENU
```

```
      WRITE (6,*) 'causes a return to MS/DOS . '
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE MENU(NMDF)
C
C  Display menu on screen
C
      INTEGER*1 ESC, TOPBAR(70), BOTBAR(70), MIDBAR(70), FRAME(2)
      INTEGER*1 SPACES(68)
      CHARACTER*30 CHOICE(8), PASSWD(8), TNAME(8), NMDF
      CHARACTER*1  PWTYPE(8), TTYPE(8)
      CHARACTER*58 EXPLAN(8), HEADER
      INTEGER*4 ITEMS
      COMMON /MVAR/ CHOICE, PASSWD, TNAME, PWTYPE,
     1              TTYPE, EXPLAN, HEADER, ITEMS
      DATA TOPBAR/-55,68*-51,-69/
      DATA BOTBAR/-56,68*-51,-68/
      DATA MIDBAR/-52,68*-51,-71/
      DATA FRAME /-70,-70/
      DATA SPACES /68*32/
      ESC  = 27
C
C Attach menu descriptor file
C
      OPEN(1,FILE=NMDF)
C
C Read its contents
C
      I = 1
      READ (1,1) HEADER
    1 FORMAT(1X,A58)
    4 READ (1,5,END=3) CHOICE(I)
    5 FORMAT(1X,A30)
      READ (1,1,END=6)        EXPLAN(I)
      READ (1,2,END=6)        PWTYPE(I), PASSWD(I)
    2 FORMAT (A1,A30)
      READ (1,2,END=6)        TTYPE(I),  TNAME(I)
      I = I + 1
      GO TO 4
    3 ITEMS = I - 1
C
C  Clear the screen
C
      WRITE (6,100) ESC, ESC
  100 FORMAT(1H&,A1,'[40m',A1,'[2J')
   50 WRITE (6,101) ESC
  101 FORMAT(1H&,A1,'[36;40m')
C
C  Set up the frame and the menu text
C
      WRITE (6,110) TOPBAR
  110 FORMAT( 5X,70A1)
      WRITE (6,110) FRAME(1), SPACES, FRAME(2)
      WRITE (6,110) MIDBAR
      DO 120 I= 1, 18
  120   WRITE (6,110) FRAME(1), SPACES, FRAME(2)
      WRITE (6,110) MIDBAR
      WRITE (6,110) FRAME(1), SPACES, FRAME(2)
```

```
      WRITE (6,110) BOTBAR
      WRITE (6,201) ESC, ESC, HEADER
  201 FORMAT (1X,A1,'[2;11H',A1,'[37;40m',A58)
      WRITE (6,202) ESC, ESC, CHOICE(1)
  202 FORMAT (1X,A1,'[5;20H',A1,'[32;40m1.  ',A30)
      DO 300 I = 2, ITEMS
      WRITE (6,203) ESC, ESC, I, CHOICE(I)
  203 FORMAT (1H&,A1,'[34D',A1,'[2B',I1,'.  ',A30)
  300 CONTINUE
      WRITE (6,204) ESC, ESC, ESC
  204 FORMAT (1H&,A1,'[21;19H',A1,'[2;34;47m',
     1'Press the key corresponding to menu choice',A1,'[37;40m')
      WRITE (6,205) ESC
  205 FORMAT (1H&,A1,'[23;7H',
     1'Press H then corresponding key for more details on menu item.')
      CLOSE (1)
      RETURN
C
C  Error trap for badly formed menu descriptor file ...
C
    6 WRITE (6,*) 'Unexpected end on menu descriptor file ',NMDF
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE CHOOSE(ITEMS, CHCE)
C
C  Get choice from menu
C
      INTEGER*4 ITEMS, CHCE
      INCLUDE 'A:SYSREG.FOR'
  100 AH = $08
      CALL SYS1(SYSREG)
      IF (AL .EQ. 72) THEN
        CHCE = 0
      ELSE
        CHCE = AL - 48
      END IF
      IF (CHCE .LT. 0 .OR. CHCE .GT. ITEMS) GO TO 100
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE HELP
C
C  Display the help line
C
      INTEGER*1 ESC
      CHARACTER*30 CHOICE(8), PASSWD(8), TNAME(8), NMDF
      CHARACTER*1  PWTYPE(8), TTYPE(8)
      CHARACTER*58 EXPLAN(8), HEADER
      INTEGER*4 ITEMS
      INCLUDE 'SYSREG.FOR'
      COMMON /MVAR/ CHOICE, PASSWD, TNAME, PWTYPE,
     1              TTYPE, EXPLAN, HEADER, ITEMS
      ESC = 27
  100 CALL CHOOSE(ITEMS, ICHCE)
      IF (ICHCE .EQ. 0) GO TO 100
      WRITE (6,110) ESC, EXPLAN(ICHCE)
```

```
  110 FORMAT (1H&,A1,'[23;7H',A58,'   ')
      WRITE (6,120) ESC
  120 FORMAT (1H&,A1,'[24;23H',
     1'Press SPACE to return to main menu')
  130 AH = $08
      CALL SYS1(SYSREG)
      IF (AL .NE. 32) GO TO 130
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE SECURE(ICHCE, OK)
C
C  Security check
C
      LOGICAL OK
      INTEGER*1 ESC
      CHARACTER*30 CHOICE(8), PASSWD(8), TNAME(8), NMDF, GUESS
      CHARACTER*1  PWTYPE(8), TTYPE(8)
      CHARACTER*58 EXPLAN(8), HEADER
      INTEGER*4 ITEMS
      COMMON /MVAR/ CHOICE, PASSWD, TNAME, PWTYPE,
     1              TTYPE, EXPLAN, HEADER, ITEMS
      ESC = 27
      IF (PWTYPE(ICHCE) .EQ. '-') THEN
C
C If no password required then all is OK
C
        OK = .TRUE.
      ELSE
C
C Otherwise ask for password
C
        WRITE (6, 100) ESC
  100   FORMAT (1H&,A1,'[23;7H',
     1'                                                                ')
        WRITE (6,110) ESC, ESC
  110   FORMAT (1H&,A1,'[23;7H','Enter password > ',A1,'[32;42m')
        READ (5,'(A)') GUESS
        CALL ENCRPT(GUESS)
        IF (GUESS .EQ. PASSWD(ICHCE)) THEN
C
C  Password correct
C
          OK = .TRUE.
          WRITE (6,140) ESC
  140     FORMAT (1H&,A1,'[37;40m')
        ELSE
C
C  Password wrong
C
          WRITE (6,120) ESC, ESC
  120     FORMAT (1H&,A1,'[23;7H',A1,'[31;47m',
     1'                        NO ACCESS                               ')
          OK = .FALSE.
          DO 130 I = 1,  300000
  130       CONTINUE
        END IF
      END IF
      RETURN
```

```
END
```

```
      PROGRAM INPUT
C
C
C
C             ********* Listing 6.3 ************
C
C                               INPUT
C                               -----
C
C       Data  input  program  for  crime  pattern analysis system ---
C  Reads a screen decription from the file in IDF; Prints this screen
C  (an input form for a single crime record) and reads in  the  data.
C  Stores  this data onto the table of beatwise crime rates, and also
C  the table of pointwise rates,  in as binary files.
C
C
C         Files :    IDF - Input Descriptor File -- Channel 1
C                  SPTCRM - Pointwise Crime Data  -- Channel 2
C                  TABCRM - Tabular   Crime Data  -- Channel 3
C
C         Other Channels
C                         - Keyed Input Of Data   -- Channel 5
C                         - Output of text to VDU -- Channel 6
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      CHARACTER IDF*20, TEMPLT*76, VERBAL*60, POST*8, INCHAR*60,
     1 SPCS*1, VARMKR*2, BLNK*78, TEXT*60
      INTEGER   INPOSN(6,2), WIDTH(6)
      LOGICAL   GIVEN(6), ALLIN, OK
      LOGICAL*1 LEAVE
      INTEGER*1 ESC, TR, BR, TL, BL, UL, AL
      DIMENSION TEMPLT(20), VARMKR(6), SPCS(78)
C
C  Above variables refer to characters to display form on VDU,
C  and to widths, codes and cursor postion for input data items.
C  QBREAK() is a test for ctrl-C --- abandon entry.
C
      INTEGER   DATEV, MONTH, YEAR, CRIME, DN, DF1, DF2, WEEK
      INTEGER   DT1, DT2, MT1, MT2, YR1, YR2, BEAT
      INTEGER   DA(16), MA(16), YA(16), CRA(16,32), DAYNUM
      INTEGER   REFNUM(16,100), NCRIMS(16), DAYNOT(16,100)
      REAL      CREAST(16,100), CRNORT(16,100)
C
C  Above variables refer to dates, beats etc. for database of crime
C  incidents
C
      EQUIVALENCE (SPCS(1), BLNK)
      DATA SPCS/78*' '/ , VARMKR/'$D','$M','$Y','$P','$T','$N'/
      DATA ESC/27/,        WIDTH/2, 2, 2, 8, 60, 4/
      DATA TR/-65/, BR/-39/, TL/-38/, BL/-64/, UL/-77/, AL/-60/
C
C  The above data statements initialise some of the constant data items
C
CCCCCCCCCCCCCCCC The Program CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Disable the break key
C
```

```
      CALL SBREAK
C
C  First, read in the Input Descriptor File into TEMPLT
C
      CALL GETCOM(IDF)
      OPEN(1, FILE=IDF)
      DO 100 I = 1, 20
  100   TEMPLT(I) = BLNK
      I = 1
  120   READ (1,'(A)', END=110) TEMPLT(I)
        I = I + 1
        IF (I.LT. 21) GO TO 120
  110 CONTINUE
      CLOSE (1)
C
C  Now check that all of the variables required have positions given
C  for input on the screen ... If not then halt the program.
C
      DO 130 I = 1, 6
        GIVEN(I) = .FALSE.
        J = 1
  135     INPOSN(I,1) = INDEX(TEMPLT(J), VARMKR(I))
          IF (INPOSN(I,1).NE.0) THEN
            INPOSN(I,2) = J
            GIVEN(I) = .TRUE.
          END IF
          J = J + 1
          IF ((.NOT.GIVEN(I)).AND.(J .LT. 21)) GO TO 135
  130     CONTINUE
      ALLIN = .TRUE.
      DO 140 I = 1, 6
  140   ALLIN = ALLIN .AND. GIVEN(I)
      IF (.NOT. ALLIN) THEN
        WRITE (6,'(A)') ' Error -- Not all variables specified in IDF'
        STOP 1
      END IF
C
C  Modify the input screen by putting dotted lines for variables
C
      DO 150 I = 1, 6
        J = INPOSN(I,1)
        K = INPOSN(I,2)
        DO 150 L = 0, WIDTH(I) - 1
  150     TEMPLT(K)(J+L:J+L) = '_'
C
C  Modify the input coordinates to allow for border
C
      DO 154 I = 1, 6
        INPOSN(I,1) = INPOSN(I,1) + 1
        INPOSN(I,2) = INPOSN(I,2) + 1
  154 CONTINUE
C
C  Read in the postcode centroid and beat information
C
      CALL SETUP
C
C  Now put the menu on the screen
C
  260 WRITE (6,180) ESC, ESC
  180 FORMAT (1X,A1,'[40m',A1,'[2J')
      WRITE (6,'(2X,78A1)') TL,(AL, I = 1, 76), TR
```

```
      DO 170 I = 1, 20
  170   WRITE (6,'(2X,A1,A76,A1)') UL, TEMPLT(I), UL
      WRITE (6,'(2X,78A1)') BL,(AL, I = 1, 76), BR
C
C  Next get the data from the user
C
C
C  Firstly get the date, month,  and year.
C
  200 CALL GETTXT(INPOSN(1,1), INPOSN(1,2), WIDTH(1), TEXT)
      IF (LEAVE()) GO TO 999
      CALL CHECK(TEXT(1:2), 1, 31, DATEV, OK)
      IF (.NOT. OK) THEN
        CALL PUTTXT(22,24,'Error in date entry: press any key')
        CALL GETKEY
        CALL PUTTXT(22,24,'                                    ')
      END IF
      IF (.NOT.OK) GO TO 200
  210 CALL GETTXT(INPOSN(2,1), INPOSN(2,2), WIDTH(2), TEXT)
      IF (LEAVE()) GO TO 999
      CALL CHECK(TEXT(1:2), 1, 12, MONTH, OK)
      IF (.NOT. OK) THEN
        CALL PUTTXT(22,24,'Error in month entry: press any key')
        CALL GETKEY
        CALL PUTTXT(22,24,'                                     ')
      END IF
      IF (.NOT. OK) GO TO 210
  220 CALL GETTXT(INPOSN(3,1), INPOSN(3,2), WIDTH(3), TEXT)
      IF (LEAVE()) GO TO 999
      CALL CHECK(TEXT(1:2), 0, 99, YEAR, OK)
      IF (.NOT. OK) THEN
        CALL PUTTXT(22,24,'Error in year entry: press any key')
        CALL GETKEY
        CALL PUTTXT(22,24,'                                    ')
      END IF
      IF (.NOT. OK) GO TO 220
C
C  Now get the postcode and see if it is a real one
C
  230   CALL GETTXT(INPOSN(4,1), INPOSN(4,2), WIDTH(4), TEXT)
        IF (LEAVE()) GO TO 999
        POST = TEXT(1:8)
        CALL LOOKUP(POST, BEAT, XREF, YREF)
        IF (BEAT .EQ. 0) THEN
          CALL PUTTXT(22,24,'Post code error: Press any key')
          CALL GETKEY
          CALL PUTTXT(22,24,'                                  ')
        END IF
        IF (BEAT .EQ. 0) GO TO 230
C
C  Now the description
C
      CALL GETTXT(INPOSN(5,1), INPOSN(5,2), WIDTH(5), TEXT)
      IF (LEAVE()) GO TO 999
      VERBAL = TEXT(1:60)
C
C  Now the crime number
C
  240 CALL GETTXT(INPOSN(6,1), INPOSN(6,2), WIDTH(6), TEXT)
      IF (LEAVE()) GO TO 999
      CALL CHECK(TEXT(1:4), 0, 9999, CRIME, OK)
```

```
      IF (.NOT. OK) THEN
        CALL PUTTXT(22,24,'Error in crime number: press any key')
        CALL GETKEY
        CALL PUTTXT(22,24,'                                      ')
      END IF
      IF (.NOT. OK) GO TO 240
C
C  Everything is now ready ... Allow user to verify the record
C
      CALL PUTTXT(22,24,'Is the above record correct (Y/N)')
  250 CALL GETTXT(56,24,1,TEXT)
      IF (INDEX('YNyn',TEXT(1:1)) .EQ. 0) THEN
        CALL PUTTXT(22,24,'Please enter Y or N                 ')
      END IF
      IF (INDEX('YNyn',TEXT(1:1)) .EQ. 0) GO TO 250
      IF (TEXT(1:1) .EQ. 'N' .OR. TEXT(1:1) .EQ. 'n') GO TO 260
C
C  Now process the data.  First make the date into a number
C
      DN = DAYNUM(DATEV, MONTH, YEAR+1900)
C
C  See if it it is sensible -- ie not in the future
C
      CALL DATE(YR2, MT2, DT2)
      DF2 = DAYNUM(DT2, MT2, YR2)
      IF (DN .GT. DF2) THEN
C
C  Trap for future dates
C
        CALL PUTTXT(15,24,'Error: Your crime is in the future!:Press any
     1 Key')
        CALL GETKEY
        CALL PUTTXT(15,24,'
     1    ')
        GO TO 999
      END IF
C
C  Now attach tabular file,  and find most recent date
C
      OPEN(3,FILE='TABCRM',FORM='UNFORMATTED')
      READ (3) YR1, MT1, DT1
      DF1 = DAYNUM(DT1, MT1, YR1)
C
C  If date is after last saturday on file, roll on a week
C
      IF (DF2 .GT. DF1) THEN
        DO 270 I = 1, 32
  270     CRA(1,I) = 0
        CALL DAMOYR((DF2/7)*7 + 7, DA(1), MA(1), YA(1))
        DF1 = DAYNUM(DA(1), MA(1), YA(1))
        DA(2) = DT1
        MA(2) = MT1
        YA(2) = YR1
        READ (3) (CRA(2,J), J = 1, 32)
        DO 280 I = 3, 16
          READ (3) YA(I), MA(I), DA(I)
  280     READ (3) (CRA(I,J), J = 1, 32)
      ELSE
        YA(1) = YR1
        MA(1) = MT1
        DA(1) = DT1
```

```
C
C  Do this if otherwise
C
        READ (3) (CRA(1,J), J = 1, 32)
        DO 290 I = 2, 16
          READ (3) YA(I), MA(I), DA(I)
  290     READ (3) (CRA(I, J), J = 1, 32)
      END IF
C
C  Update the tabular records
C
      WEEK = (DF1 - DN) / 7 + 1
      IF (WEEK .LE. 16) CRA(WEEK,BEAT) = CRA(WEEK,BEAT) + 1
C
C  Overwrite the old file
C
      CLOSE (3)
      CALL CMND('COPY TABCRM EMERG.TAB >X')
      CALL CMND('ERASE TABCRM')
      OPEN(3, FILE = 'TABCRM', FORM = 'UNFORMATTED')
      DO 300 I = 1, 16
        WRITE (3) YA(I), MA(I), DA(I)
  300   WRITE (3) (CRA(I, J), J = 1, 32)
      CLOSE (3)
C
C  Now update the points file -- Re read YR1, MT1, DT1 to skip on file
C
      OPEN (2, FILE = 'SPTCRM', FORM = 'UNFORMATTED')
      READ (2) YR1, MT1, DT1
      DF1 = DAYNUM(DT1, MT1, YR1)
C
C  If todays date exceeds last saturday, roll it on a week
C  DF2 already known from Tabular data
C
      IF (DF2 .GT. DF1) THEN
        NCRIMS(1) = 0
        CALL DAMOYR((DF2/7)*7 + 7, DA(1), MA(1), YA(1))
        DF1 = DAYNUM(DA(1), MA(1), YA(1))
        DA(2) = DT1
        MA(2) = MT1
        YA(2) = YR1
        READ (2) NCRIMS(2)
        READ (2) (CREAST(2,J), J=1, 100)
        READ (2) (CRNORT(2,J), J=1, 100)
        READ (2) (DAYNOT(2,J), J=1, 100)
        READ (2) (REFNUM(2,J), J=1, 100)
        DO 330 I = 3, 16
          READ (2) YA(I), MA(I), DA(I)
          READ (2) NCRIMS(I)
          READ (2) (CREAST(I,J), J=1, 100)
          READ (2) (CRNORT(I,J), J=1, 100)
          READ (2) (DAYNOT(I,J), J=1, 100)
  330     READ (2) (REFNUM(I,J), J=1, 100)
      ELSE
          YA(1) = YR1
          MA(1) = MT1
          DA(1) = DT1
          READ (2) NCRIMS(1)
          READ (2) (CREAST(1,J), J=1, 100)
          READ (2) (CRNORT(1,J), J=1, 100)
          READ (2) (DAYNOT(1,J), J=1, 100)
```

```
          READ (2) (REFNUM(1,J), J=1, 100)
        DO 340 I = 2, 16
          READ (2) YA(I), MA(I), DA(I)
          READ (2) NCRIMS(I)
          READ (2) (CREAST(I,J), J=1, 100)
          READ (2) (CRNORT(I,J), J=1, 100)
          READ (2) (DAYNOT(I,J), J=1, 100)
  340     READ (2) (REFNUM(I,J), J=1, 100)
      END IF
C
C  Update point-based records
C
      WEEK = (DF1 - DN) / 7 + 1
      IF (WEEK .LE. 16) THEN
        NCRIMS(WEEK) = NCRIMS(WEEK) + 1
        CREAST(WEEK, NCRIMS(WEEK)) = XREF / 100.0
        CRNORT(WEEK, NCRIMS(WEEK)) = YREF / 100.0
        DAYNOT(WEEK, NCRIMS(WEEK)) = DN
        REFNUM(WEEK, NCRIMS(WEEK)) = CRIME
      END IF
C
C  Overwrite the old file
C
      CLOSE (2)
      CALL CMND('COPY SPTCRM EMERG.SPT >X')
      CALL CMND('ERASE SPTCRM')
      OPEN (2, FILE='SPTCRM', FORM = 'UNFORMATTED')
      DO 350 I = 1, 16
        WRITE (2) YA(I), MA(I), DA(I)
        WRITE (2) NCRIMS(I)
        WRITE (2) (CREAST(I,J), J=1, 100)
        WRITE (2) (CRNORT(I,J), J=1, 100)
        WRITE (2) (DAYNOT(I,J), J=1, 100)
  350   WRITE (2) (REFNUM(I,J), J=1, 100)
      CLOSE (2)
C
C  Check performance against machine prediction
C
      CALL EXECPG('MONITOR', IFAULT)
C
C Download text and crime record number
C
      OPEN (8, FILE='APPREC')
      WRITE (8,'(I8,A60)') CRIME, VERBAL
      CLOSE (8)
      CALL CMND('COPY TXTCRM+APPREC >x')
      CALL CMND('ERASE APPREC')
C
C  Go round again if needed
C
  999 CALL PUTTXT(22,24,'Do you have any further crime records? (Y/N)')
  310 CALL GETTXT(74,24,1,TEXT)
      IF (INDEX('YNyn',TEXT(1:1)) .EQ. 0) THEN
        CALL PUTTXT(22,24,'Please enter Y or N                ')
      END IF
      IF (INDEX('YNyn',TEXT(1:1)) .EQ. 0) GO TO 310
      IF (TEXT(1:1) .EQ. 'Y' .OR. TEXT(1:1) .EQ. 'y') GO TO 260
      CALL RBREAK
      STOP
      END
C
```

```
C*********************************************************************
C
      SUBROUTINE GETTXT(XCR, YCR, WIDTH, TEXT)
      INTEGER*4 XCR, YCR, WIDTH, POSN
      CHARACTER*(*) TEXT
      CHARACTER*1   LETTER
      INTEGER*1 SYSREG(20), AL, AH, BL, BH, CL, CH, DL, DH
      LOGICAL*1 ZF, CF
      INTEGER*2 BP, SI, DI, DS, ES, AX, BX, CX, DX
      EQUIVALENCE  (ZF, SYSREG(1)), (CF, SYSREG(2)),
     1             (BP, SYSREG(3)),
     2             (SI, SYSREG(5)), (DI, SYSREG(7)),
     3             (DS, SYSREG(8)), (ES, SYSREG(11)),
     4             (AX, AL, SYSREG(13)), (AH, SYSREG(14)),
     5             (BX, BL, SYSREG(15)), (BH, SYSREG(16)),
     6             (CX, CL, SYSREG(17)), (CH, SYSREG(18)),
     7             (DX, DL, SYSREG(19)), (DH, SYSREG(20))
      EQUIVALENCE (LETTER, AL)
C
C  Subroutine to put a cursor onto a given point on the vdu
C  and read a string of width WIDTH.
C
C
C  Initially the text to return is filled with spaces
C
      DO 100 I = 1, WIDTH
  100   TEXT(I:I) = ' '
C
C  Put the cursor in position
C
      AH = 2
      BH = 0
      DL = XCR
      DH = YCR
      CALL SYS2(16,SYSREG)
C
C  Now read the input ... do not allow cursor off pro-forma
C
      POSN = 1
  110 AH = 8
      CALL SYS1(SYSREG)
      IF (AL .GE. 32) THEN
C
C  Non - control character handling
C
        IF (POSN .LE. WIDTH) THEN
          WRITE (6,'(1H&,A1)') LETTER
          TEXT(POSN:POSN) = LETTER
          POSN = POSN + 1
        END IF
      ELSE
C
C  Special characters --- return,  backspace, forespace
C

        IF (AL .EQ. 8 .OR. AL .EQ. 11) THEN
C                                  Backspace
          IF (POSN .GT. 1) THEN
             POSN = POSN - 1
             WRITE (6,'(1H&,A1)') LETTER
          END IF
```

```
        END IF
        IF (AL .EQ. 26) THEN
C                                  Forespace
          IF (POSN .LT. WIDTH) THEN
             WRITE (6,'(1H&,A1)') LETTER
             POSN = POSN + 1
          END IF
        END IF
C
C  End of loop --- re-loop if return has not been pressed
C
      END IF
      IF (AL .NE. 13) GO TO 110
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE CHECK(STRING, LOWER, UPPER, NUMBER, FLAG)
C
C  Subroutine to detect errors in strings of numbers
C
C
C  STRING Text string to be scanned
C  UPPER  Upper bound for number
C  LOWER  Lower bound for number
C  NUMBER value returned if number good
C  FLAG   Logical test of whether number good
C
      CHARACTER*(*) STRING
      CHARACTER*4   ARGMNT
      INTEGER*4 UPPER, LOWER, NUMBER
      LOGICAL FLAG
C
C  Ensure ARGMNT has four characters
C
      IF (LEN(STRING) .LT. 4) THEN
        ARGMNT = '  '//STRING
      ELSE
        ARGMNT = STRING
      END IF
C
C  Try to read the string as a number.  If OK check range
C
        READ (ARGMNT,'(I4)',ERR=100) NUMBER
        FLAG = NUMBER .GE. LOWER .AND. NUMBER .LE. UPPER
        RETURN
C
C  If program gets here,  text was not a number
C
  100   FLAG = .FALSE.
        RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE PUTTXT(XTL, YTL, TEXT)
C
C  Subroutine to write text on screen
C
C  XTL              X Text Location
```

```
C  YTL               Y Text Location
C  TEXT              The text
C
      INTEGER*4 XTL, YTL
      CHARACTER*(*) TEXT
      INTEGER*1 SYSREG(20), AL, AH, BL, BH, CL, CH, DL, DH
      LOGICAL*1 ZF, CF
      INTEGER*2 BP, SI, DI, DS, ES, AX, BX, CX, DX
      EQUIVALENCE  (ZF, SYSREG(1)), (CF, SYSREG(2)),
     1             (BP, SYSREG(3)),
     2             (SI, SYSREG(5)), (DI, SYSREG(7)),
     3             (DS, SYSREG(8)), (ES, SYSREG(11)),
     4             (AX, AL, SYSREG(13)), (AH, SYSREG(14)),
     5             (BX, BL, SYSREG(15)), (BH, SYSREG(16)),
     6             (CX, CL, SYSREG(17)), (CH, SYSREG(18)),
     7             (DX, DL, SYSREG(19)), (DH, SYSREG(20))
      AH = 2
      BH = 0
      DL = XTL
      DH = YTL
      CALL SYS2(16,SYSREG)
      WRITE (6,'(1H&,A)') TEXT
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE GETKEY
C
C  Wait for a key to be pressed
C
      INTEGER*1 SYSREG(20), AL, AH, BL, BH, CL, CH, DL, DH
      LOGICAL*1 ZF, CF
      INTEGER*2 BP, SI, DI, DS, ES, AX, BX, CX, DX
      EQUIVALENCE  (ZF, SYSREG(1)), (CF, SYSREG(2)),
     1             (BP, SYSREG(3)),
     2             (SI, SYSREG(5)), (DI, SYSREG(7)),
     3             (DS, SYSREG(8)), (ES, SYSREG(11)),
     4             (AX, AL, SYSREG(13)), (AH, SYSREG(14)),
     5             (BX, BL, SYSREG(15)), (BH, SYSREG(16)),
     6             (CX, CL, SYSREG(17)), (CH, SYSREG(18)),
     7             (DX, DL, SYSREG(19)), (DH, SYSREG(20))
      AH = 8
      CALL SYS1(SYSREG)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE LOOKUP(POST, BEAT, XREF, YREF)
      CHARACTER*8 POST, SQUASH
      INTEGER*4 BEAT, INDEX1, INDEX2
      REAL*4 XREF, YREF
C
C  Lookup table from Postcode to coordinates and beat
C
C  POST Postcode
C  BEAT Integer beat code
C  XREF, YREF 10 metre grid coordinates
C
C
```

```
C  First,  remove spaces from postcode
C
      SQUASH = '        '
      INDEX2 = 1
      DO 100 INDEX1 = 1, 8
        IF (POST(INDEX1:INDEX1) .NE. ' ') THEN
          SQUASH(INDEX2:INDEX2) = POST(INDEX1:INDEX1)
          INDEX2 = INDEX2 + 1
        END IF
  100 CONTINUE
      CALL BINLUT(SQUASH(1:7), BEAT, XREF, YREF)
      RETURN
      END
C
C*********************************************************************
C
      INTEGER*4 FUNCTION DAYNUM(DATE, MONTH, YEAR)
C
C  Gives an integer corresponding to the date
C  allowing dates to be subtracted etc.
C
      INTEGER*4 YEAR, MONTH, DATE
      IF (MONTH .LT. 3) THEN
        DAYNUM = 365*YEAR + DATE + 31*(MONTH - 1) + (YEAR - 1)/4
     1         - INT(0.75*((YEAR - 1)/100) + 1)
      ELSE
        DAYNUM = 365*YEAR + DATE + 31*(MONTH - 1) - INT(0.4*MONTH + 2.3)
     1         + YEAR/4 - INT(0.75*(YEAR/100) + 1)
      END IF
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE DAMOYR(NUMBER, DATE, MONTH, YEAR)
C
C  Opposite of DAYNUM --- given the number gives date, month, year
C
      INTEGER*4 NUMBER, DATE, MONTH, YEAR, GUESS, TEST
C
C  Initial guess at the year -- normally correct
C
      YEAR = NUMBER / 365.24
      GUESS = NUMBER - 365*YEAR - (YEAR - 1)/4 + 15
C
C  If incorrect,  the previous year will work
C
      IF (GUESS .LE. 0) THEN
        YEAR = YEAR - 1
        GUESS = NUMBER - 365*YEAR - (YEAR - 1)/4   + 15
      END IF
C
C  Now find the month -- Jan & Feb first,  then the rest
C
      IF (GUESS .LT. 32) THEN
        DATE = GUESS
        MONTH = 1
        RETURN
      END IF
      IF (GUESS .LT. 60) THEN
        DATE = GUESS - 31
```

```
        MONTH = 2
        IF (DATE .EQ. 29) THEN
          IF ((YEAR/4)*4 .NE. YEAR) THEN
            MONTH = 3
            DATE = 1
          END IF
        END IF
        RETURN
      END IF
      IF (GUESS .GE. 60) THEN
        MONTH = 2
  100   MONTH = MONTH + 1
        TEST = GUESS - 31*(MONTH - 1) + (0.4*MONTH + 2.3)
        IF (TEST .GT. 0) GO TO 100
      END IF
      MONTH = MONTH - 1
C
C   What remains gives the data of the month
C
      DATE = GUESS - 31*(MONTH - 1) + (0.4*MONTH + 2.3)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE SETUP
C
C   The Initialiser for the Look-Up table routine
C
      CHARACTER*7 CODE(1564)
      INTEGER*4 BEAT(1564)
      REAL X(1564), Y(1564)
      COMMON /LOOK/ CODE, BEAT, X, Y
      OPEN(4,FILE='B:POSTCOD.SQZ',FORM='UNFORMATTED')
      READ (4) CODE
      READ (4) BEAT
      READ (4) X
      READ (4) Y
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE BINLUT(CODE, BEAT, X, Y)
C
C  The Action Routine for the lookup table
C
      CHARACTER CODE*7
      CHARACTER*7 TCODE(1564)
      INTEGER*4 TBEAT(1564), BEAT
      REAL TX(1564), TY(1564), X, Y
      INTEGER LOOKH, LOOKL, LOOKM, GAP, LGAP
      LOGICAL FOUND, LLT
      COMMON /LOOK/ TCODE, TBEAT, TX, TY
C
C Start the search by initialising the parameters
C
      LOOKH = 1564
      LOOKL = 1
      FOUND = .FALSE.
      LGAP = 2*(LOOKH - LOOKL)
```

```
      GAP  = LOOKH - LOOKL
C
C Main loop of a binary division search
C
  100 IF ((GAP .NE. LGAP).AND.(.NOT.FOUND)) THEN
        LOOKM = (LOOKH + LOOKL) / 2
        FOUND = TCODE(LOOKM).EQ.CODE
        IF (.NOT.FOUND) THEN
          IF (LLT(TCODE(LOOKM),CODE)) THEN
            LOOKL = LOOKM
          ELSE
            LOOKH = LOOKM
          END IF
          LGAP = GAP
          GAP  = LOOKH - LOOKL
        END IF
      GO TO 100
      END IF
C
C  Result of search may now be transferred
C
      IF (FOUND) THEN
        X = TX(LOOKM)
        Y = TY(LOOKM)
        BEAT = TBEAT(LOOKM)
      ELSE
        BEAT = 0
      END IF
      RETURN
      END
C
C*********************************************************************
C
      LOGICAL*1 FUNCTION LEAVE()
      LOGICAL*1 QBREAK
      CHARACTER*1 TEXT
C
C  Device to handle user interrupts to exit record
C
      LEAVE = QBREAK()
C
C  Get verification of this
C
      IF (LEAVE) THEN
        CALL PUTTXT(21,24,'Request to exit record -- Verify (Y/N)')
  100   CALL GETTXT(65,24,1,TEXT)
        CALL PUTTXT(21,24,'                                                ')
        IF (INDEX('YNyn',TEXT(1:1)) .EQ. 0) THEN
          CALL PUTTXT(22,24,'Please enter Y or N                           ')
          GO TO 100
        END IF
        LEAVE = (TEXT(1:1) .EQ. 'Y' .OR. TEXT(1:1) .EQ. 'y')
      END IF
      RETURN
      END
```

```
      PROGRAM RASTER
C
C     ********** Listing 6.4 **********
C
C
C
C
C                 Rasterising Program  -  turns a GIMMS
C                  outline into a crushed raster code
C
C
C                 Uses a point in polygon routine
C
C                 Channels 1= Gimms Outline
C                          2= Rasterised output
C                          5= Limits of map
C
C
C
      INTEGER*2 GRID(375,300), LAST, COUNT, GR2(375,300)
      REAL*4 XOUTL(400),  YOUTL(400), XTOP, XBTM, YTOP, YBTM
      REAL*4 XLT, XLB, YLT, YLB, XSTEP, YSTEP
      INTEGER*4 SIZE, IL, IH, JL, JH, DELTX, DELTY
      INTEGER*4 LIMS(4,32)
      INTEGER*2 MINSCN
      CHARACTER*4 BEAT
      LOGICAL WITHIN
C
C  Input the limit points of the map from file
C
      READ (5,'(4F7.0)') XBTM, XTOP, YBTM, YTOP
C
C  Compute the steps for centroids of pixels (in terms of National grid)
C
      XSTEP = (XTOP - XBTM) / 375.0
      YSTEP = (YTOP - YBTM) / 300.0
      DO 887 I = 1, 375
        DO 887 J = 1, 300
  887     GRID(I,J) = 0
C
C  Read the beats in from GIMMS area dump file
C
      DO 888 IBT = 1,  32
      READ (1,'(T10,A4)') BEAT
      READ (1,'(T10,I4)') SIZE
      READ (1,'(10F7.0)') (XOUTL(I), YOUTL(I), I = 1, SIZE)
      WRITE (6,'(2X,A4)') BEAT
C
C  Compute the x and y limits for each zone
C
      XLT = XOUTL(1)
      XLB = XOUTL(1)
      YLT = YOUTL(1)
      YLB = YOUTL(1)
      DO 100 I = 2, SIZE
        IF (XLT. LT. XOUTL(I)) XLT = XOUTL(I)
        IF (XLB. GT. XOUTL(I)) XLB = XOUTL(I)
        IF (YLT. LT. YOUTL(I)) YLT = YOUTL(I)
        IF (YLB. GT. YOUTL(I)) YLB = YOUTL(I)
  100 CONTINUE
      IL = INT((XLB - XBTM)/XSTEP) - 2
```

```
      IH = INT((XLT - XBTM)/XSTEP) + 2
      JL = INT((YLB - YBTM)/YSTEP) - 2
      JH = INT((YLT - YBTM)/YSTEP) + 2
      LIMS(1,IBT) = IL
      LIMS(2,IBT) = IH
      LIMS(3,IBT) = JL
      LIMS(4,IBT) = JH
      IF (IL .LE. 0) IL = 1
      IF (IH .GE. 375) IH = 375
      IF (JL .LE. 0) JL = 1
      IF (IL .GE. 300) JH = 300
C
C  Check,  for each zone,  whether centroid is inside, foe each
C  centroid in the x and y limit square
C
      DO 110 I = IL, IH
        DO 110 J = JL, JH
          CALL INSIDE(XOUTL, YOUTL, SIZE,
     1                XBTM + (I - 0.5)*XSTEP, YBTM + (J - 0.5)*YSTEP,
     2                WITHIN)
          IF (WITHIN) GRID(I,J) = IBT
  110   CONTINUE
C
C  Output it in run-length encoded form (see text)
C
      CALL CRUNCH(GRID,IL,JL,IH,JH,IBT)
  888 CONTINUE
C
C  Now do edge detection;  for finding borders
C
      DO 889 IBT = 1, 32
        IL = LIMS(1,IBT)
        IH = LIMS(2,IBT)
        JL = LIMS(3,IBT)
        JH = LIMS(4,IBT)
        DO 890 I = IL + 1,  IH - 1
          DO 890 J = JL + 1, JH - 1
C
C  Look around each pixel for neighbours of a different zone code
C  When one is found less than zone code of zone whose borders are
C  being detected,  then the pixel is a border.  Otherwise not.
C
C
            IF (GRID(I,J) .EQ. IBT) THEN
            MINSCN = 32000
            IF (GRID(I  ,J-1).LT.MINSCN) MINSCN = GRID(I  ,J-1)
            IF (GRID(I-1,J  ).LT.MINSCN) MINSCN = GRID(I-1,J  )
            IF (GRID(I+1,J  ).LT.MINSCN) MINSCN = GRID(I+1,J  )
            IF (GRID(I  ,J+1).LT.MINSCN) MINSCN = GRID(I  ,J+1)
            IF (MINSCN .LT. GRID(I,J)) THEN
              GR2(I,J) = 1
            ELSE
              GR2(I,J) = 0
            END IF
            ELSE
             GR2(I,J) = 0
            END IF
  890       CONTINUE
        WRITE (6,'(A)') '++++++++++++++++++++'
C
C  Run length encode the border pixels in the same way as the area ones
```

```
C
       CALL CRUNCH(GR2,IL+1,JL+1,IH-1,JH-1,1)
  889  CONTINUE
       STOP
       END
C
C*************************************************************************
C
       SUBROUTINE INSIDE(XOUTL,YOUTL,LEN,XREF,YREF,WITHIN)
C
C  Subroutine taken from Baxter, 1976 (see refs)  to detect
C  whether a point lies inside a vector defined polygon
C
       INTEGER*4 LEN, COUNT, PTR
       REAL*4    XOUTL(LEN), YOUTL(LEN), XREF, YREF
       LOGICAL*4 WITHIN
       J = 0
       WITHIN = .TRUE.
C
C  Code from here taken directly from text :  no structuring!
C
       DO 11 I=2, LEN
       M = 0
       IF ((YOUTL(I-1)-YREF)*(YREF-YOUTL(I))) 11, 5, 9
     5 IF (YOUTL(I-1)-YOUTL(I)) 8, 6, 7
     6 IF ((XOUTL(I-1)-XREF)*(XREF-XOUTL(I))) 11, 12, 12
     7 M = M - 2
     8 M = M - 1
     9 M = M + 2
       IF ((YREF-YOUTL(I-1))*(XOUTL(I)-XOUTL(I-1))/
      1 (YOUTL(I)-YOUTL(I-1))+XOUTL(I-1) - XREF) 11, 12, 10
    10 J = J + M
    11 CONTINUE
C
C  Final test result
C
       WITHIN = J/4*4.NE.J
    12 RETURN
       END
C
C*************************************************************************
C
       SUBROUTINE CRUNCH(GRID,IL,JL,IH,JH,IBT)
C
C  Run length encode the data in matrix grid
C  Find number of zeroes, number of ones etc .. until end of
C  horizontal line
C
       INTEGER*2 GRID(375,300), COUNT
C
C  Top left hand corner of zone coordinates on screen (Mode 16 EGA)
C
       WRITE (6,*) IL, JL
C
C  Scan horizontal line
C
       DO 130 J = JL, JH
C
C Does it start inside or outside of zone
C
         IF (GRID(IL,J).EQ.IBT)  THEN
```

```
          INIBT = 1
        ELSE
          INIBT = 0
        END IF
        WRITE (6,*) INIBT
C
C Scan along run until state of inside/outside changes
C
        LAST = INIBT
        COUNT = 1
        DO 140 I = IL+1, IH
          IF (GRID(I,J).EQ.IBT)  THEN
            INIBT = 1
          ELSE
            INIBT = 0
          END IF
          IF (INIBT.EQ.LAST) THEN
            COUNT = COUNT + 1
          ELSE
            WRITE (6,*) COUNT
            COUNT = 1
            LAST = 1 - LAST
          END IF
  140     CONTINUE
C
C end of line / end of scan information
C
        IF (LAST .EQ. 1) WRITE (6,*) COUNT
        IF (J.NE.JH) WRITE (6,*) 0
        IF (J.EQ.JH) WRITE (6,*) -1
  130   CONTINUE
      RETURN
      END
```

```
      PROGRAM CRUNCH
C
C        ******** Listing 6.5 ********
C
C     Takes the ASCII run length encoded map description file
C     generated by MTS (Listing 6.4) and converts it to a binary
C     file readable by MS-DOS,  which is faster to access from the
C     graphics  programs
C
      CHARACTER   NAME*4, DFILE*30
      INTEGER*2 ARRAY(1800)
      INTEGER*4 NPTS
      LOGICAL   BORDER
C
C  What file is the data in?
C
      CALL GETCOM(DFILE)
C
C  Is it a border file (in thise case filenane contains a + )
C
      ILOC = INDEX(DFILE,'+')
      IF (ILOC .NE. 0) THEN
        BORDER = .TRUE.
        DFILE(ILOC:ILOC) = ' '
      ELSE
        BORDER = .FALSE.
      END IF
C
C  Access the file to put the binary data into
C
      OPEN (1, FILE=DFILE,FORM='UNFORMATTED')
C
C  Main crunching loop
C
  120 CONTINUE
C
C  Report zone name (= BORD for a border file)
C
      IF (BORDER) THEN
        NAME = 'BORD'
      ELSE
        READ (5,'(2X,A4)', END =110) NAME
      END IF
C
C  Get the zone info (as in Listing 6.4)
C
      NPTS = 3
      READ (5,*,END = 110) ARRAY(1), ARRAY(2)
  100 READ (5,*) ARRAY(NPTS)
        NPTS = NPTS + 1
        IF (ARRAY(NPTS - 1) .NE. -1) GO TO 100
      NPTS = NPTS - 1
C
C  Output it in binary form
C
      WRITE (1) NAME,NPTS,(ARRAY(K), K = 1, NPTS)
      WRITE (6,'(A)') ' Zone '//NAME//' crunched.'
      GO  TO 120
C
C  Loop ends here
C
```

```
110 CLOSE(1)
    STOP
    END
```

```
C
C
C       ******** Listing 6.6 ********
C
C
C
C
C
C
      SUBROUTINE ZONE(ZARRAY, COL, PITCH)
C
C  Subroutine to shade a compacted zone array with given
C  COLour and PITCH
C
      INTEGER*2 ZARRAY(800)
      INTEGER*4 COL, PITCH, I, J, EDGEX, APTR, STATE, MOVE
      EDGEX = ZARRAY(1)
C
C  Convert to screen coordinates
C
      J = 341 - ZARRAY(2)
      I = EDGEX
      APTR = 2
  130 APTR = APTR + 1
      STATE = ZARRAY(APTR)
  120 APTR = APTR + 1
      MOVE =  ZARRAY(APTR)
C
C  Scan through each horizontal line of data,  using MOD and the PITCH
C  value to see if each pixel is supposed to be illuminated
C  If a run length for scan of less than zero is encountered
C  then exit loop
C
      IF (MOVE .LE. 0) GO TO 100
      IF (STATE .EQ. 0) THEN
        I = I + MOVE
        STATE = 1 - STATE
      ELSE
        DO 110 K = I, I + MOVE - 1
          MASK = 1 - MIN(MOD(K+J,PITCH),1)
  110     CALL DOT(K,J,MASK*COL)
        I = I + MOVE
        STATE = 1 - STATE
      END IF
      GO TO 120
C
C On a second -1 jump out of zone drawing routine
C
  100 IF (MOVE .EQ. -1) RETURN
      I = EDGEX
      J = J - 1
      GO TO 130
      END
```

```
C
C
C       ******** Listing 6.7 ********
C
C
      SUBROUTINE MAP(REGION,PVALS,CVALS,NZONES)
C
C  Draws a map given a region file (see text) and two arrays
C  for Pitch,  and colour of the region (PVALS & CVALS)
C
      CHARACTER*(*) REGION
      INTEGER*4 PTR,  PITCH, COL, NZONES, PVALS(NZONES), CVALS(NZONES)
      INTEGER*2 SHAPE(1800)
      CHARACTER*4 NAME
C
C Open a region file (a set of ZONES to shade)
C
      OPEN (1, FILE=REGION, FORM='UNFORMATTED')
C
C Shade each zone in turn
C
      DO 100 IBT = 1, NZONES
        READ (1) NAME,PTR,(SHAPE(I),I=1,PTR)
        PITCH = PVALS(IBT)
        COL   = CVALS(IBT)
        IF (PITCH .NE. 0) CALL ZONE(SHAPE, COL, PITCH)
  100 CONTINUE
      CLOSE(1)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE ZONE(ZARRAY, COL, PITCH)
C
C  Subroutine to shade a compacted zone array with given
C  COLour and PITCH
C
      INTEGER*2 ZARRAY(800)
      INTEGER*4 COL, PITCH, I, J, EDGEX, APTR, STATE, MOVE
      EDGEX = ZARRAY(1)
C
C  Convert to screen coordinates
C
      J = 341 - ZARRAY(2)
      I = EDGEX
      APTR = 2
  130 APTR = APTR + 1
      STATE = ZARRAY(APTR)
  120 APTR = APTR + 1
      MOVE =  ZARRAY(APTR)
C
C  Scan through each horizontal line of data,  using MOD and the PITCH
C  value to see if each pixel is supposed to be illuminated
C  If a run length for scan of less than zero is encountered
C  then exit loop
C
      IF (MOVE .LE. 0) GO TO 100
      IF (STATE .EQ. 0) THEN
        I = I + MOVE
        STATE = 1 - STATE
```

```
      ELSE
        DO 110 K = I, I + MOVE - 1
          MASK = 1 - MIN(MOD(K+J,PITCH),1)
  110     CALL DOT(K,J,MASK*COL)
        I = I + MOVE
        STATE = 1 - STATE
      END IF
      GO TO 120
C
C On a second -1 jump out of zone drawing routine
C
  100 IF (MOVE .EQ. -1) RETURN
      I = EDGEX
      J = J - 1
      GO TO 130
      END
C
C*************************************************************************
C
      SUBROUTINE MODE(N)
C
C Set graphics mode
C
      INCLUDE 'A:SYSREG.FOR'
      AH = 0
      AL = N
      CALL SYS2(16, SYSREG)
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE DOT(I, J, COL)
C
C  Plot a pixel at I,J of colour COL
C
      INCLUDE 'A:SYSREG.FOR'
      INTEGER COL
      AH = $0C
      BH = 0
      CX = I
      DX = J
      AL = COL
      CALL SYS2(16,SYSREG)
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE PUTTXT(XTL, YTL, TEXT)
C
C  Put TEXT at position XTL, YTL
C
      INTEGER*4 XTL, YTL
      CHARACTER*(*) TEXT
      INCLUDE 'A:SYSREG.FOR'
      AH = 2
      BH = 0
      DL = XTL
      DH = YTL
      CALL SYS2(16,SYSREG)
```

```
      WRITE (6,'(1H&,A)') TEXT
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE BOXES(COL)
      INTEGER*4 COL
C
C  Draw boxes on screen for crime mapping system
C
      DO 100 I = 0, 639
        CALL DOT(I,0,COL)
  100   CALL DOT(I,349,COL)
      DO 110 J = 1, 348
        CALL DOT(0,J,COL)
        CALL DOT(639,J,COL)
  110   CALL DOT(455,J,COL)
      RETURN
      END
```

```
      PROGRAM CHORO
C
C              ****** Listing 6.8 *******
C
C      Choropleth mapping of past data
C
C
      INTEGER*4 PVALS(32), CVALS(32), CRIMES(16,32),
     1          YR(16), MO(16), DA(16), WK, MARRAY(32)
      CHARACTER*1 DUMMY
      CHARACTER*8 PERIOD
      CHARACTER*10 DATETX
      REAL*4 CVEC(32), UPPER(32), PRED(32)
      INCLUDE 'A:SYSREG.FOR'
C
C Access the data file for beatwise crime counts
C
      OPEN (3,FILE='TABCRM',FORM='UNFORMATTED')
      DO 50 I = 1, 16
        READ (3) YR(I), MO(I), DA(I)
   50   READ (3) (CRIMES(I,J), J = 1, 32)
      CLOSE (3)
C
C Put up map title and control panel
C
      WK = 1
      PERIOD = ' 7 Days '
      WRITE (DATETX, '(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
      CALL MODE(16)
      CALL BOXES(15)
      DO 99  I = 1, 32
   99   MARRAY(I) = CRIMES(WK,I)
      CALL PUTTXT(20,1,'South Gosforth Subdivision')
      CALL PUTTXT(22,2,'Household Burglaries')
      CALL CHKEY(1)
      CALL CHKEY(2)
      CALL PUTTXT(58, 2,'     Menu :-')
      CALL PUTTXT(58, 4,'<+> = advance 1 wk')
      CALL PUTTXT(58, 6,'<-> = go back 1 wk')
      CALL PUTTXT(58, 8,'<4> = 4 weeks data')
      CALL PUTTXT(58,10,'<8> = 8 weeks data')
      CALL PUTTXT(58,12,'</> =16 weeks data')
      CALL PUTTXT(58,14,'<H> = High Risk   ')
      CALL PUTTXT(58,16,'<E> = Exit to Menu')
   51 CALL PUTTXT(19,3,PERIOD//'Ending '//DATETX)
C
C  Overlay blank maps
C
      DO 105 I = 1, 32
  105   CVALS(I) = 16
      CALL MAP('BEATS',PVALS,CVALS,32)
C
C  Now overlay final maps: code zones according to count in
C  current time period
C
      DO 100 I = 1, 32
        CVALS(I) = 1
        IPVAL    = MARRAY(I)
        PVALS(I) = 1
        IF (IPVAL .LT. 8) PVALS(I) = 2
        IF (IPVAL .LT. 6) PVALS(I) = 4
```

```
        IF (IPVAL .LT. 4) PVALS(I) = 8     390
        IF (IPVAL .LT. 2) PVALS(I) = 0
  100   CONTINUE
      CALL MAP('BEATS',PVALS,CVALS,32)
C
C  Insert the border values and plot borders
C
      DO 110 I = 1, 32
        CVALS(I) = 1
        PVALS(I) = 1
  110   CONTINUE
      CALL MAP('BORDERS',PVALS,CVALS,32)
C
C  Main menu loop
C
  500 AH = $08
      CALL SYS1(SYSREG)
      ICHCE = AL
      IF (ICHCE .EQ. 43) THEN
C
C  Go back one week and draw map (on pressing -)
C
        WK = WK - 1
        IF (WK .EQ. 0) WK = 1
        DO 510 I = 1, 32
  510     MARRAY(I) = CRIMES(WK, I)
        PERIOD = ' 7 Days '
        WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
        GO TO 51
      END IF
      IF (ICHCE .EQ. 45) THEN
C
C  Go forward one week and draw map (On pressing +)
C
        WK = WK + 1
        IF (WK .EQ. 17) WK = 16
        DO 520 I = 1, 32
  520     MARRAY(I) = CRIMES(WK, I)
        PERIOD = ' 7 Days '
        WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
        GO TO 51
      END IF
      IF (ICHCE .EQ. 52) THEN
C
C  Change map to drawing over a 4-week period (on pressing 4)
C
        PERIOD = '4 Weeks '
        WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(1), MO(1), YR(1)
        DO 530 I = 1, 32
          MARRAY(I) = CRIMES(1,I) + CRIMES(2,I) + CRIMES(3,I)
          MARRAY(I) = MARRAY(I)   + CRIMES(4,I)
  530     MARRAY(I) = MARRAY(I) / 4.0
        GO TO 51
      END IF
      IF (ICHCE .EQ. 56) THEN
C
C  Change map to an 8-week period (on pressing 8)
C
        PERIOD = '8 Weeks '
        WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(1), MO(1), YR(1)
        DO 540 I = 1, 32
```

```
          MARRAY(I) = 0
          DO 550 J = 1, 8
  550       MARRAY(I) = MARRAY(I) + CRIMES(J,I)
  540     MARRAY(I) = MARRAY(I) / 8.0
        GO TO 51
      END IF
      IF (ICHCE .EQ. 56) THEN
C
C  Change map to a 16 week period (on pressing /)
C
        PERIOD = '16 Week '
        WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(1), MO(1), YR(1)
        DO 560 I = 1, 32
          MARRAY(I) = 0
          DO 570 J = 1, 16
  570       MARRAY(I) = MARRAY(I) + CRIMES(J,I)
  560     MARRAY(I) = MARRAY(I) / 16.0
        GO TO 51
      END IF
      IF (ICHCE .EQ. 72) THEN
C
C  Highlight 'High Risk beats' (on pressing H)
C
        DO 600 I = 1, 32
  600     CVEC(I) = CRIMES(WK,I)
        CALL RISK(CVEC, CVALS, 32)
        CALL MAP('BEATS',PVALS,CVALS,32)
        DO 620 I = 1, 32
  620     CVALS(I) = 15
        CALL MAP('C:BORDERS',PVALS,CVALS,32)
        GO TO 500
      END IF
      IF (ICHCE .NE. 69) GO TO 500
C
C  If keypress is not E then re-loop;  else exit menu
C
      CALL MODE(3)
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE CHKEY(NKEY)
C
C  Key routine for choropleth maps
C
      IF (NKEY .EQ. 1) THEN
        CALL KEYBOX(45, 5,'0 -< 2',1, 0)
        CALL KEYBOX(45, 7,'2 -< 4',1, 8)
        CALL KEYBOX(45, 9,'4 -< 6',1, 4)
        CALL KEYBOX(45,11,'6 -< 8',1, 2)
        CALL KEYBOX(45,13,'8 -<  ',1, 1)
        CALL PUTTXT(40,18,'Crimes per Week')
      END IF
      IF (NKEY .EQ. 2) THEN
        CALL KEYBOX(45,16,'High Risk',12, 1)
      END IF
      RETURN
      END
C
C*********************************************************************
```

```
C
      SUBROUTINE KEYBOX(XTL, YTL, TEXT, COL, PITCH)
C
C  Draw box of a given pitch and colour next to text at point
C  XTL, YTL on mode 16 EGA.
C
      INTEGER*4 XTL, YTL, COL, PITCH, GTX, GTY, PCOL
      CHARACTER*(*) TEXT
      GTX = XTL*8   - 20
      GTY = YTL*14  -  4
      IF (PITCH .NE. 0) THEN
        DO 100 I = GTX, GTX + 15
          DO 100 J = GTY, GTY + 21
            IF (MOD(I+J,PITCH) .EQ. 0) THEN
              PCOL = COL
            ELSE
              PCOL = 0
            END IF
            CALL DOT(I,J,PCOL)
  100       CONTINUE
      END IF
      DO 110 I = GTX,  GTX + 15
        CALL DOT(I,GTY,15)
  110   CALL DOT(I,GTY+21,15)
      DO 120 J = GTY,  GTY + 21
        CALL DOT(GTX,J,15)
  120   CALL DOT(GTX+15,J,15)
      CALL PUTTXT(XTL, YTL, TEXT)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE RISK(CVEC, CVALS, NZONES)
C
C  Subroutine to identify high risk beats (ie those exceeding
C  predicted values
C
      INTEGER*4 CVEC(NZONES), CVALS(NZONES), FLAG(32)
C
C  Set all colours to blank initially
C
      DO 100 I = 1, NZONES
  100   CVALS(I) = 0
C
C   Use MONITOR to find high risk beats
C
      OPEN (10, FILE='COMP.MON')
      READ (10) FLAG
      CLOSE (10)
      DO 110 I = 1, NZONES
        IF (FLAG .NE. 0) CVALS(I) = 12
  110 CONTINUE
      RETURN
      END
```

```
      PROGRAM PNTMAP
C
C           ******* Listing 6.9 ******
C
C  Program to plot point incidence of crime
C
C
      INTEGER*4 DIDS(16,100), YR(16), MO(16), DA(16), WK, NPTS(16)
      INTEGER*4 PVALS(32), CVALS(32), ENDWK, CINTV, REFNUM(16,100)
C
C  Data required includes standard crime database (pointwise section)
C  and area/boundary data for the subdivision
C
C
      REAL*4    XPTS(16,100), YPTS(16,100)
      CHARACTER*1 DUMMY
      CHARACTER*10 DATETX, BDATE
      LOGICAL OVRLAY, PLOTTD(100)
      INCLUDE 'A:SYSREG.FOR'
C
C  Initialise variables
C
      DATA PVALS/32*1/
      DATA CVALS/32*16/
      OVRLAY = .FALSE.
C
C  Get defaults for Knox cluster definition (Critical time and distance)
C
      OPEN (8,FILE='KNOX.BIN',FORM='UNFORMATTED')
      READ (8) CDIST, CINTV
      CDIST = (CDIST/100.0) ** 2
      CLOSE (8)
C
C  Open the crime database (for Points)
C
      OPEN (18, FILE='SPTCRM', FORM='UNFORMATTED')
C
C  For past 16 weeks
C
      DO 50 I = 1, 16
C
C  Find date for end of week
C
        READ (18) YR(I), MO(I), DA(I)
C
C  Number of incidents in week
C
        READ (18) NPTS(I)
C
C  and day, grid ref., and police ref no. of each incident
C
        READ (18) (XPTS(I,J),J=1,100)
        READ (18) (YPTS(I,J),J=1,100)
        READ (18) (DIDS(I,J),J=1,100)
50      READ (18) (REFNUM(I,J),J=1,100)
      CLOSE (18)
      WK = 1
      ENDWK = 1
C
C  Now put up the control display surrounding the map
C
```

```
      WRITE (DATETX, '(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
      CALL MODE(16)
      CALL BOXES(15)
      CALL PUTTXT(20,1,'South Gosforth Subdivision')
      CALL PUTTXT(22,2,'Household Burglaries')
      CALL PUTTXT(58, 2,'     Menu :-')
      CALL PUTTXT(58, 4,'<+> = advance 1 wk')
      CALL PUTTXT(58, 6,'<-> = go back 1 wk')
      CALL PUTTXT(58, 8,'<O> = Overlay On  ')
      CALL PUTTXT(58,10,'<C> = Clusters    ')
      CALL PUTTXT(58,12,'<S> = Select Event')
      CALL PUTTXT(58,14,'<E> = Exit to Menu')
      CALL PTKEY
C
C  Start of main menu loop : repeat until E is pressed
C
   51 CONTINUE
      IF (OVRLAY) THEN
        CALL PUTTXT(20,3,BDATE//' to '//DATETX)
      ELSE
        CALL PUTTXT(20,3,'7 Days Ending '//DATETX)
      END IF
C
C  When not in OVERLAY mode,  erase the currents points on VDU
C  if there are any
C
      IF (.NOT.OVRLAY) THEN
        DO 52 I = 1, 16
          DO 53 J = 1, NPTS(I)
   53       CALL POINT(XPTS(I,J),YPTS(I,J),4180.,5650.,120.,120.,8)
   52     CONTINUE
C
C  Then plot the map borders,  if they got damaged above
C
        DO 110 I = 1, 32
          CVALS(I) = 15
          PVALS(I) = 1
  110     CONTINUE
        CALL MAP('C:BORDERS',PVALS,CVALS,32)
      END IF
C
C  Plot the current weeks data
C
      DO 100 I = 1, NPTS(WK)
        CALL POINT(XPTS(WK,I), YPTS(WK,I), 4180., 5650., 120., 120., 2)
  100   CONTINUE
C
C  Main Menu Loop ... Get Key
C
  500 AH = $08
      CALL SYS1(SYSREG)
      ICHCE = AL
C
C  Go forward a week ( After pressing + )
C
      IF (ICHCE .EQ. 43) THEN
        IF (.NOT. OVRLAY) THEN
          WK = WK - 1
          IF (WK .EQ. 0) WK = 1
          WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
          GO TO 51
```

```
        END IF
        GO TO 500
      END IF
C
C  Go back a week ( After pressing - )
C
      IF (ICHCE .EQ. 45) THEN
        WK = WK + 1
        IF (WK .EQ. 17) WK = 16
        IF (OVRLAY) THEN
          WRITE (BDATE ,'(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
        ELSE
          WRITE (DATETX,'(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
        END IF
        GO TO 51
      END IF
C
C  Toggle switch on OVERLAY mode ( After pressing O )
C
      IF (ICHCE .EQ. 79) THEN
        OVRLAY = .NOT. OVRLAY
        IF (OVRLAY) THEN
          CALL PUTTXT(71, 8,' Off')
        ELSE
          CALL PUTTXT(71, 8,' On ')
        END IF
      END IF
C
C  Scan for Knox clusters ( After pressing C )
C
      IF (ICHCE .EQ. 67 .AND. NPTS(WK) .GT. 1) THEN
        IF (.NOT. OVRLAY) THEN
          DO 92 I = 1, 16
            DO 93 J = 1, NPTS(I) - 1
   93         CALL POINT(XPTS(I,J),YPTS(I,J),4180.,5650.,120.,120.,8)
   92       CONTINUE
        DO  75 I = 1, 32
          CVALS(I) = 15
          PVALS(I) = 1
   75     CONTINUE
        CALL MAP('C:BORDERS',PVALS,CVALS,32)
        END IF
        DO 900 I = 1, NPTS(WK)
  900     PLOTTD(I) = .FALSE.
        DO 910 I = 1, NPTS(WK)
          IF (.NOT.PLOTTD(I))
     1      CALL SCAN(I, NPTS, XPTS, YPTS, DIDS, PLOTTD, WK,CDIST,CINTV,
     2      PLOTTD)
  910     CONTINUE
      END IF
C
C  Select a point to view comment text ( After pressing S )
C
      IF (ICHCE .EQ. 83) THEN
        CALL SELECT(NPTS, XPTS, YPTS, REFNUM, WK,
     1  4180., 5650., 120., 120.)
      END IF
C
C  If E not pressed then default (do nothing)
C
C  If E pressed then exit the case structure
```

```
C
      IF (ICHCE .NE. 69) GO TO 500
      CALL MODE(3)
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE PTKEY
C
C  Print the key to the point map on the VDU
C
C
        CALL KEYPNT(45, 9,'1 Event',12,2)
        CALL KEYPNT(45,11,'2 Events',13,2)
        CALL KEYPNT(45,13,'3+',14,2)
        CALL KEYPNT(45,15,'Cluster',10,4)
        CALL PUTTXT(40,18,'Crime Locations')
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE KEYPNT(XTL, YTL, TEXT, COL, SYMTYP)
      INTEGER*4 XTL, YTL, COL, GTX, GTY, PCOL, SYMTYP
C
C   Print the text next to the key on the map
C
      CHARACTER*(*) TEXT
      GTX = XTL*8  - 8
      GTY = YTL*14 +  7
      CALL MARK(GTX, GTY, COL, SYMTYP)
      CALL PUTTXT(XTL, YTL, TEXT)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE POINT(X,Y,XTL,YTL,XWD,YWD,SYMTYP)
C
C  Put a point on the map,  given the National Grid cornerpoints
C  and the point coordinates in National Grid scale.
C
      REAL*4     X, Y, XTL, YTL, XWD, YWD
      INTEGER*4  I, J, COL, SYMTYP, BLKSYM
C
C  Convert to 'screen coordinates' in EGA mode 16
C
      I = INT(((X - XTL) / XWD) * 375.0)
      J = INT(((Y - YTL) / YWD) * 300.0)
      J = 341 - J
C
C  Plot it with an appropriate symbol
C
      IF (SYMTYP .LE. 4) THEN
C
C  Cumulative colouring for multiple occurrence of same pixel
C
        CALL GDOT(I, J, COL)
        IF (COL .EQ. 0 .OR. COL .EQ. 15) CALL MARK(I, J, 12, SYMTYP)
        IF (COL .EQ. 12) CALL MARK(I, J, 13, SYMTYP)
```

```
        IF (COL .EQ. 13) CALL MARK(I, J, 14, SYMTYP)
      END IF
      IF (SYMTYP .GT. 4 .AND. SYMTYP .LE. 8) THEN
C
C  Blackout : for clearing points
C
        BLKSYM = SYMTYP - 4
        CALL MARK(I, J, 16, BLKSYM)
      END IF
      IF (SYMTYP .GT. 8) THEN
C
C  For cluster symbol
C
        BLKSYM = SYMTYP - 8
        CALL MARK(I, J,10, SYMTYP)
      END IF
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE GDOT(I, J, COL)
C
C Find out what COLour dat at position I,J is
C
      INCLUDE 'A:SYSREG.FOR'
      INTEGER COL
      AH = $0D
      BH = 0
      CX = I
      DX = J
      CALL SYS2(16,SYSREG)
      COL = AL
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE MARK(I, J, COL, SYMTYP)
      INTEGER I, J, COL, SYMTYP
C
C Plot the marker symbol
C
      CALL DOT(I, J, COL)
      IF (SYMTYP.NE.1) THEN
        IF (SYMTYP.NE.3) THEN
C
C  Cross - shaped for crime incidence
C
          CALL DOT(I-1,J,COL)
          CALL DOT(I+1,J,COL)
          CALL DOT(I,J-1,COL)
          CALL DOT(I,J+1,COL)
        END IF
        IF (SYMTYP.NE.2) THEN
C
C  Square  shaped for Cluster identification
C
          CALL DOT(I-1,J-1,COL)
          CALL DOT(I+1,J-1,COL)
          CALL DOT(I-1,J+1,COL)
```

```
          CALL DOT(I+1,J+1,COL)
        END IF
      END IF
      RETURN
      END
      SUBROUTINE SCAN(I, NPTS, XPTS, YPTS, DIDS, PLOTTD,WK,CDIST, CINTV,
     1 PLOTTD)
C
C  Scan for Knox clusters
C
      INTEGER*4 NPTS(16), DIDS(16,100), WK, WKPTR, DAYGAP, CINTV
      REAL*4    XPTS(16,100), YPTS(16,100), DIST, CDIST
      LOGICAL   PLOTTD(100)
C
C  Scan through previous week (if available) : plot if necessary
C
      WKPTR = WK + 1
      IF (WKPTR .LT. 17 .AND. NPTS(WKPTR) .GT. 0) THEN
        J = 1
  100     DAYGAP = IABS(DIDS(WKPTR,J) - DIDS(WK, I))
          IF (DAYGAP .LE. CINTV) THEN
            DIST = (XPTS(WKPTR,J)-XPTS(WK,I))**2 +
     1             (YPTS(WKPTR,J)-YPTS(WK,I))**2
            IF (DIST .LE. CDIST) THEN
              CALL POINT(XPTS(WK,I),YPTS(WK,I),4180.,5650.,120.,120.,12)
              PLOTTD(I) = .TRUE.
              RETURN
            END IF
          END IF
          J = J + 1
          IF (J .LE. NPTS(WKPTR)) GO TO 100
      END IF
C
C  Scan through this week : plot if necessary
C
      J = I + 1
  110 IF (.NOT.PLOTTD(J)) THEN
        DAYGAP = IABS(DIDS(WK,J) - DIDS(WK, I))
        IF (DAYGAP .LE. CINTV) THEN
          DIST = (XPTS(WK,J)-XPTS(WK,I))**2 +
     1           (YPTS(WK,J)-YPTS(WK,I))**2
          IF (DIST .LE. CDIST) THEN
            CALL POINT(XPTS(WK,I),YPTS(WK,I),4180.,5650.,120.,120.,12)
            PLOTTD(I) = .TRUE.
            RETURN
          END IF
        END IF
        J = J + 1
        IF (J .LE. NPTS(WK)) GO TO 110
      END IF
C
C  Scan through next week (if there is one)  : plot if necessary
C
      WKPTR = WK - 1
      IF (WKPTR .GT. 0 .AND. NPTS(WKPTR) .GT. 0) THEN
        J = 1
  120     DAYGAP = IABS(DIDS(WKPTR,J) - DIDS(WK, I))
          IF (DAYGAP .LE. CINTV) THEN
            DIST = (XPTS(WKPTR,J)-XPTS(WK,I))**2 +
     1             (YPTS(WKPTR,J)-YPTS(WK,I))**2
            IF (DIST .LE. CDIST) THEN
```

```
              CALL POINT(XPTS(WK,I),YPTS(WK,I),4180.,5650.,120.,120.,12)
              PLOTTD(I) = .TRUE.
              RETURN
            END IF
          END IF
          J = J + 1
          IF (J .LE. NPTS(WKPTR)) GO TO 120
        END IF
        RETURN
        END
C
C*****************************************************************
C
        SUBROUTINE SELECT(NPTS, XPTS, YPTS,REFNUM, WK, XTL, YTL, XWD, YWD)
C
C  Select an incident --- try to find verbal description
C
        REAL*4 XPTS(16,100), YPTS(16,100), XTL, YTL, XWD, YWD
        CHARACTER CNTEXT*4, DESCR*60
        INTEGER NPTS(16), REFNUM(16,100), WK, CNUM, ENV(12), CCREF
        LOGICAL DISTBD
        INCLUDE 'SYSREG.FOR'
        CNUM = 1
        DISTBD = .FALSE.
        WRITE (CNTEXT,'(I4)') REFNUM(WK,CNUM)
        CALL OUTLIN(ENV, XPTS(WK,CNUM), YPTS(WK,CNUM),
       1 XTL, YTL, XWD, YWD)
C
C  Menu of options
C
        IF (NPTS(WK) .GT. 0) THEN
          CALL PUTTXT(58,21,'Ref. No: '//CNTEXT)
          CALL PUTTXT(58,16,'<Z> Select last')
          CALL PUTTXT(58,17,'<X> Select next')
          CALL PUTTXT(58,18,'<V> View Comment')
          CALL PUTTXT(58,19,'<M> Main Map')
  100     AH = $08
          CALL SYS1(SYSREG)
          ICHCE = AL
C
C If border frame on screen disturbed,  then set it right again
C
          IF (DISTBD) THEN
            CALL PUTTXT(10,24,
     1  '                                                                ')
            DO 105 I = 0, 639
  105         CALL DOT(I, 349, 15)
            DO 107 I = 334, 348
  107         CALL DOT(455, I, 15)
            DISTBD = .FALSE.
          END IF
          CALL RESTOR(ENV, XPTS(WK,CNUM), YPTS(WK,CNUM),
       1   XTL, YTL, XWD, YWD)
C
C  Go back one crime incidence  (if key pressed is Z )
C
          IF (ICHCE .EQ. 90) THEN
            CNUM = CNUM - 1
            IF (CNUM .EQ. 0) CNUM = 1
C
C  Mark current incident on map
```

```
C
          CALL OUTLIN(ENV, XPTS(WK,CNUM), YPTS(WK,CNUM),
     1    XTL, YTL, XWD, YWD)
          WRITE (CNTEXT,'(I4)') REFNUM(WK,CNUM)
          CALL PUTTXT(58,21,'Ref. No: '//CNTEXT)
        END IF
C
C  Go Forward one crime incidence (if key pressed is X )
C
        IF (ICHCE .EQ. 88) THEN
          CNUM = CNUM + 1
          IF (CNUM .GT. NPTS(WK)) CNUM = NPTS(WK)
C
C  Mark current incidence on map
C
          CALL OUTLIN(ENV, XPTS(WK,CNUM), YPTS(WK,CNUM),
     1    XTL, YTL, XWD, YWD)
          WRITE (CNTEXT,'(I4)') REFNUM(WK,CNUM)
          CALL PUTTXT(58,21,'Ref. No: '//CNTEXT)
        END IF
        IF (ICHCE .EQ. 86) THEN
C
C View the comment from the database ( If V key pressed )
C
          OPEN (8, FILE='TXTCRM')
C
C Find the reference number in the text file
C
  120     READ (8,'(I8,A60)') CCREF, DESCR
            IF (CCREF.NE.REFNUM(WK,CNUM)) GO TO 120
          CALL PUTTXT(10,24,DESCR)
          CALL OUTLIN(ENV, XPTS(WK,CNUM), YPTS(WK,CNUM),
     1    XTL, YTL, XWD, YWD)
          WRITE (CNTEXT,'(I4)') REFNUM(WK,CNUM)
          CALL PUTTXT(58,21,'Ref. No: '//CNTEXT)
          CLOSE (8)
          DISTBD = .TRUE.
        END IF
        IF (ICHCE .NE. 77) GO TO 100
C
C If M not pressed (for return to map) then loop to menu read
C Otherwise exit
C
        CALL PUTTXT(58,16,'               ')
        CALL PUTTXT(58,17,'               ')
        CALL PUTTXT(58,18,'               ')
        CALL PUTTXT(58,19,'               ')
        CALL PUTTXT(58,21,'               ')
      ELSE
C
C If no crimes occur in the selected time period
C
        CALL PUTTXT(58,16,'No items')
        CALL PUTTXT(58,17,'Press SPACE  ')
  110   AH = $08
        CALL SYS1(SYSREG)
        ICHCE = AL
        IF (ICHCE .NE. 32) GO TO 110
        CALL PUTTXT(58,16,'        ')
        CALL PUTTXT(58,17,'             ')
      END IF
```

```
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE OUTLIN(ENV, X, Y, XTL, YTL, XWD, YWD)
      REAL*4 X, Y, XTL, XWD, YTL, YWD
      INTEGER ENV(12)
C
C  Put an outline around the selected point at x,y
C
C
C  Find the pixel equivalent of x, y (call it i,j)
C
      I = INT(((X - XTL) / XWD) * 375.0)
      J = INT(((Y - YTL) / YWD) * 300.0)
      J = 341 - J
C
C  Store what is currently there
C
      CALL GDOT(I-1,J  ,ENV(1))
      CALL GDOT(I-1,J+1,ENV(2))
      CALL GDOT(I  ,J+1,ENV(3))
      CALL GDOT(I+1,J+1,ENV(4))
      CALL GDOT(I+1,J  ,ENV(5))
      CALL GDOT(I+1,J-1,ENV(6))
      CALL GDOT(I  ,J-1,ENV(7))
      CALL GDOT(I-1,J-1,ENV(8))
      CALL GDOT(I-2,J  ,ENV(9))
      CALL GDOT(I+2,J  ,ENV(10))
      CALL GDOT(I  ,J+2,ENV(11))
      CALL GDOT(I  ,J-2,ENV(12))
C
C  Put a ring around it
C
      CALL DOT(I-1,J  ,11)
      CALL DOT(I-1,J+1,11)
      CALL DOT(I  ,J+1,11)
      CALL DOT(I+1,J+1,11)
      CALL DOT(I+1,J  ,11)
      CALL DOT(I+1,J-1,11)
      CALL DOT(I  ,J-1,11)
      CALL DOT(I-1,J-1,11)
      CALL DOT(I-2,J  ,11)
      CALL DOT(I+2,J  ,11)
      CALL DOT(I  ,J+2,11)
      CALL DOT(I  ,J-2,11)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE RESTOR(ENV, X, Y, XTL, YTL, XWD, YWD)
      REAL*4 X, Y, XTL, XWD, YTL, YWD
      INTEGER ENV(12)
C
C  Remove outline around the selected point at x,y
C
C
C  Find the pixel equivalent of x, y (call it i,j)
C
```

```
      I = INT(((X - XTL) / XWD) * 375.0)
      J = INT(((Y - YTL) / YWD) * 300.0)
      J = 341 - J
C
C  Put back the old contents
C
      CALL DOT(I-1,J  ,ENV(1))
      CALL DOT(I-1,J+1,ENV(2))
      CALL DOT(I  ,J+1,ENV(3))
      CALL DOT(I+1,J+1,ENV(4))
      CALL DOT(I+1,J  ,ENV(5))
      CALL DOT(I+1,J-1,ENV(6))
      CALL DOT(I  ,J-1,ENV(7))
      CALL DOT(I-1,J-1,ENV(8))
      CALL DOT(I-2,J  ,ENV(9))
      CALL DOT(I+2,J  ,ENV(10))
      CALL DOT(I  ,J+2,ENV(11))
      CALL DOT(I  ,J-2,ENV(12))
      RETURN
      END
```

```
      PROGRAM GFUNCT
C
C  ********* Listing  6.10 ************
C
C  create the kernel function for listing 6.11 as a grid of point
C  estimates
C
C
      INTEGER*2 GFUN(0:10,0:8), YORD, LORD, COL
      CHARACTER*1 ANS
C
C  Display shapes of kernel that are offered
C
    1 CALL MODE(16)
      CALL PUTTXT(31,2,'Select Kernel Type')
      CALL PUTTXT(25,5,'1.  Conic')
      CALL PUTTXT(25,9,'2.  Parabolic')
      CALL PUTTXT(25,13,'3.  Exponential')
      CALL PUTTXT(25,17,'4.  Gaussian')
      DO 100 I = 0, 41
        CALL DOT(336,266-I,15)
        CALL DOT(336,210-I,15)
        CALL DOT(336,154-I,15)
        CALL DOT(336, 98-I,15)
  100 CONTINUE
      DO 110 I = 0, 47
        CALL DOT(336+I,266,15)
        CALL DOT(336+I,210,15)
        CALL DOT(336+I,154,15)
        CALL DOT(336+I, 98,15)
  110 CONTINUE
C
C  Linear curve
C
      LORD = 41
      DO 120 I = 1, 47
        YORD = 41.0 - (I*1.3)
        IF (YORD.GT.0) THEN
          DO 130 J = YORD, LORD
            CALL DOT(336+I, 98-J,11)
  130       CONTINUE
        END IF
      LORD = YORD
  120 CONTINUE
C
C  Parabolic curve
C
      LORD = 41
      DO 140 I = 1, 47
        YORD = 41.0 - (I*I*0.033)
        IF (YORD.GT.0) THEN
          DO 150 J = YORD, LORD
            CALL DOT(336+I,154-J,12)
  150       CONTINUE
        END IF
      LORD = YORD
  140 CONTINUE
      LORD = 41
C
C  Exponential curve
C
```

```
      DO 160 I = 1, 47
        YORD = 41.0 * EXP(-I/10.0)
        IF (YORD.GT.0) THEN
          DO 170 J = YORD, LORD
            CALL DOT(336+I,210-J,13)
  170       CONTINUE
        END IF
      LORD = YORD
  160 CONTINUE
C
C  Gaussian curve
C
      LORD = 41
      DO 180 I = 1, 47
        YORD = 41.0 * EXP(-I*I/350.0)
        IF (YORD.GT.0) THEN
          DO 190 J = YORD, LORD
            CALL DOT(336+I,266-J,14)
  190       CONTINUE
        END IF
      LORD = YORD
  180 CONTINUE
C
C  Choose kernel shape and bandwidth
C
      CALL PUTTXT(30,20,'Kernel Code (1-4) > ')
      READ (5,*) KSHAPE
      CALL PUTTXT(30,22,'Enter Bandwidth   > ')
      READ (5,*) BW
      BW = BW / 8.0
      CALL PUTTXT(30,24,'Please Wait .....   ')
C
C  Assign distances to grid points in first quadrant (rest are done
C  in kernel program by symmetry
C
      IF (KSHAPE .EQ. 1) THEN
C
C  Conic surface (from linear decay)
C
        A = 50.0 / BW
        DO 200 I = 0, 10
          DO 200 J = 0, 8
            DIST = I*I*16 + J*J*25
  200       GFUN(I,J) = IFIX(100.0 - A*SQRT(DIST))
      END IF
      IF (KSHAPE .EQ. 2) THEN
C
C  Parabolic Surface
C
        A = 50.0 / BW**2
        DO 210 I = 0, 10
          DO 210 J = 0, 8
            DIST = I*I*16 + J*J*25
  210       GFUN(I,J) = IFIX(100.0 - A*DIST)
      END IF
      IF (KSHAPE .EQ. 3) THEN
C
C  Exponential surface
C
        A = 0.693147 / BW
        DO 220 I = 0, 10
```

```
          DO 220 J = 0, 8
            DIST = I*I*16 + J*J*25
  220       GFUN(I,J) = IFIX(100.0*EXP(-A*SQRT(DIST)))
        END IF
        IF (KSHAPE .EQ. 4) THEN
C
C  Gaussian surface
C
          A = 0.693147 / BW**2
          DO 230 I = 0, 10
            DO 230 J = 0, 8
              DIST = I*I*16 + J*J*25
  230         GFUN(I,J) = IFIX(100.0*EXP(-A*DIST))
        END IF
        DO 240 I = 0, 10
          DO 240 J = 0, 8
            IF (GFUN(I,J).LT.0)  GFUN(I,J) = 0
  240       CONTINUE
C
C  Draw a colour coded picture to show what shape GFUN is
C
        CALL MODE(16)
        CALL PUTTXT(22,2,'Kernel Function Around Single Point')
        DO 250 I = 0, 10
          DO 250 J = 0, 8
            COL = GFUN(I,J) / 10
            COL = COL + 1
            DO 260 IX = 0, 9
              DO 260 IY = 0, 9
                CALL DOT(315+I*10+IX,171+J*10+IY,COL)
                IF (I.GT.0) THEN
                  CALL DOT(315-I*10+IX,171+J*10+IY,COL)
                  IF (J.GT.0) THEN
                    CALL DOT(315-I*10+IX,171-J*10+IY,COL)
                  END IF
                END IF
                IF (J.GT.0) THEN
                  CALL DOT(315+I*10+IX,171-J*10+IY,COL)
                END IF
  260       CONTINUE
  250   CONTINUE
C
C  Draw a scale for the picture just drawn
C
        DO 270 I = 0, 109
          COL = I / 10 + 1
          DO 270 J = 269, 278
            CALL DOT(260+I, J, COL)
  270       CONTINUE
        CALL PUTTXT(23,19,'Low Risk')
        CALL PUTTXT(48,19,'High Risk')
        CALL PUTTXT(36,20,'Scale')
C
C  If user likes this,  enter it on the file
C  If not - compute a new GFUN
C
C
        CALL PUTTXT(25,22,'Commit this to file (Y/N) ? >')
        READ (5,'(A)') ANS
        IF (ANS .EQ. 'Y' .OR. ANS .EQ. 'y') THEN
          OPEN(7,FILE='KERNEL.BIN',FORM='UNFORMATTED')
```

```
  WRITE (7) GFUN
ELSE
  GO TO 1
END IF
CALL MODE(3)
STOP
END
```

```
      PROGRAM KERMAP
C
C           ******* Listing 6.11 *******
C
C  Kernel estimation program.  Draws shaded contour map on screen.
C
C
      INTEGER*4 DIDS(16,100), YR(16), MO(16), DA(16), WK, NPTS(16)
      INTEGER*4 PVALS(32), CVALS(32), ENDWK, GVALS(32), REFNUM(16,100)
C
C  Accesses crime database (point section):  Builds up
C  a matrix of risks in KERN,  with Kernel function GFUN
C
      INTEGER*2 KERN(-2:378,-2:303), GFUN(0:10,0:8)
      REAL*4    XPTS(16,100), YPTS(16,100)
      CHARACTER*1 DUMMY
      CHARACTER*10 DATETX, BDATE
      LOGICAL OVRLAY, PLOTTD(100)
      INCLUDE 'A:SYSREG.FOR'
C
C  Initialise values
C
      DATA PVALS/32*1/
      DATA CVALS/32*15/
      DATA GVALS/32*2/
C
C  Access point crime database
C
      OPEN (2, FILE='SPTCRM', FORM='UNFORMATTED')
      DO 50 I = 1, 16
        READ (2) YR(I), MO(I), DA(I)
        READ (2) NPTS(I)
        READ (2) (XPTS(I,J),J=1,100)
        READ (2) (YPTS(I,J),J=1,100)
        READ (2) (DIDS(I,J),J=1,100)
   50   READ (2) (REFNUM(I,J),J=1,100)
C
C  Variables are as in listing 6.9
C
C
C  Access the binary matrix representation of the Kernel function
C
      OPEN (7,FILE='KERNEL.BIN',FORM='UNFORMATTED')
      READ (7) GFUN
C
C Set up the display
C
      WK = 1
      ENDWK = 1
      WRITE (DATETX, '(I2,1H/,I2,1H/,I4)') DA(WK), MO(WK), YR(WK)
      CALL MODE(16)
      CALL BOXES(15)
      CALL PUTTXT(20, 1,'South Gosforth Subdivision')
      CALL PUTTXT(21, 2,'Household Burglary Risk')
      CALL PUTTXT(58, 8,'<B> = Beat Zones  ')
      CALL PUTTXT(58,10,'<Z> = Risk Zones  ')
      CALL PUTTXT(58,12,'<E> = Exit to Menu')
      CALL KEYBOX(45, 7,' Low  Risk',2,1)
      CALL KEYBOX(45,10,' Med. Risk',14,1)
      CALL KEYBOX(45,13,' High Risk',4,1)
C
```

```
C  Main Menu Loop begins here
C
   51 CONTINUE
      CALL PUTTXT(20,3,'16 Wks Ending '//DATETX)
C
C  Initialise the kernel estimate by setting to zero:
C
C  It might be slow,  so put a 'Please Weight' message
C
      CALL PUTTXT(15,7,'Data Analysis : Stage 1')
      CALL KWIPE(KERN)
C
C  Begin the estimation process : put up a second 'Please Wait' .
C
      CALL PUTTXT(15,7,'Data Analysis : Stage 2')
      DO 100 WK = 1, 16
        DO 100 I = 1, NPTS(WK)
        CALL KREG(XPTS(WK,I), YPTS(WK,I),
     1  4180., 5650., 120., 120.,KERN,GFUN)
  100   CONTINUE
      CALL PUTTXT(15,7,'                          ')
C
C  Display the result as a 3 - colour contour map
C
      CALL  MAP('B:BEATS',PVALS,GVALS,32)
      CALL  MAP('B:BORDERS',PVALS,CVALS,32)
      CALL CMAP('B:BEATS',KERN,32)
C
C  Await keypress
C
  500 AH = $08
      CALL SYS1(SYSREG)
      ICHCE = AL
C
C  Overlay beat boundaries (On pressing B)
C
      IF (ICHCE .EQ. 66) THEN
        CALL MAP('B:BORDERS',PVALS,CVALS,32)
        GO TO 500
      END IF
      IF (ICHCE .EQ. 90) THEN
C
C  Overlay risk zones (ie Contours) (On pressing Z)
C
        CALL CMAP('B:BEATS',KERN,32)
        GO TO 500
      END IF
      IF (ICHCE .NE. 69) GO TO 500
C
C  If E not pressed await next keystroke
C
C
C  Otherwise exit menu section
C
      CALL MODE(3)
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE KREG(X,Y,XTL,YTL,XWD,YWD,KERN,GFUN)
```

```
C
C  Updates the kernel estimator (in screen coordinates) when given
C  a new point (in National Grid coordinates)
C
      REAL*4     X, Y, XTL, YTL, XWD, YWD
      INTEGER*4  I, J, COL, IG, JG, XTARG, YTARG
      INTEGER*2  KERN(-2:378,-2:303), GFUN(0:10,0:8)
C
C  Perform the conversion to screen coordinates
C
      I = INT(((X - XTL) / XWD) * 375.0)
      J = INT(((Y - YTL) / YWD) * 300.0)
C
C  Update the kernel estimator.   Use 4 way symmetry to reduce
C  storage overheads.  Also note that kernel is in integer form
C  to speed up computation
C
      DO 100 IG = 0, 10
        DO 100 JG = 0, 8
          XTARG = I + IG
          YTARG = J + JG
          IF (XTARG.LE.375.AND.YTARG.LE.300)
     1      KERN(XTARG,YTARG) = KERN(XTARG,YTARG) + GFUN(IG,JG)
          XTARG = I - IG
          IF (XTARG.GT.0.AND.YTARG.LE.300.AND.IG.GT.0)
     1      KERN(XTARG,YTARG) = KERN(XTARG,YTARG) + GFUN(IG,JG)
          YTARG = J - JG
          IF (XTARG.GT.0.AND.YTARG.GT.0.AND.IG.GT.0.AND.JG.GT.0)
     1      KERN(XTARG,YTARG) = KERN(XTARG,YTARG) + GFUN(IG,JG)
          XTARG = I + IG
          IF (XTARG.LE.375.AND.YTARG.GT.0.AND.JG.GT.0)
     1      KERN(XTARG,YTARG) = KERN(XTARG,YTARG) + GFUN(IG,JG)
  100 CONTINUE
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE CZONE(ZARRAY, KERN)
C
C  Similar to ZONE but plots contours within zones
C
      INTEGER*2 ZARRAY(800), KERN(-2:378,-2:303)
      INTEGER*4 COL, PITCH, I, J, EDGEX, APTR, STATE, MOVE
      EDGEX = ZARRAY(1)
      J = 341 - ZARRAY(2)
      I = EDGEX
      APTR = 2
  130 APTR = APTR + 1
      STATE = ZARRAY(APTR)
  120 APTR = APTR + 1
      MOVE =  ZARRAY(APTR)
      IF (MOVE .LE. 0) GO TO 100
      IF (STATE .EQ. 0) THEN
        I = I + MOVE
        STATE = 1 - STATE
      ELSE
        DO 110 K = I, I + MOVE - 1
C
C  Instead of the usual filter using MOD here,  a 3-stage
C  classification of the value in the kernel grid controls
```

```
C  which colour the pixel is illuminated.
C  Red = High   Yellow = Medium   Green = Low   Risk
C
          IF (KERN(K,341-J).GT. 30) CALL DOT(K,J,14)
          IF (KERN(K,341-J).GT.150) CALL DOT(K,J,4)
  110     CONTINUE
        I = I + MOVE
        STATE = 1 - STATE
      END IF
      GO TO 120
  100 IF (MOVE .EQ. -1) RETURN
      I = EDGEX
      J = J - 1
      GO TO 130
      END
C
C*********************************************************************
C
      SUBROUTINE CMAP(REGION,KERN,NZONES)
C
C  Provides a contour map for each record (corresponding to a zone)
C  in the file REGION
C
      CHARACTER*(*) REGION
      INTEGER*4 PTR,  PITCH, COL, NZONES
      INTEGER*2 SHAPE(1800), KERN(-2:303,-2:378)
      CHARACTER*4 NAME
C
C  Attach region file
C
      OPEN (1, FILE=REGION, FORM='UNFORMATTED')
C
C  Output a contoured zone for each region
C
      DO 100 IBT = 1, NZONES
        READ (1) NAME,PTR,(SHAPE(I),I=1,PTR)
        CALL CZONE(SHAPE, KERN)
  100 CONTINUE
      CLOSE(1)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE KWIPE(KERN)
C
C Set kernel estimation matrix to all zero
C
      INTEGER*2 KERN(-2:378,-2:303)
      DO 100 I = -2, 378
        DO 100 J = -2, 303
  100     KERN(I,J) = 0
      RETURN
      END
```

```
      PROGRAM PRDMAP
C
C             *********  Listing 6.12 **********
C
C
C  Bayesian Prediction program
C
C
      INTEGER*4 PVALS(32), CVALS(32), CRIMES(16,32), PY, PM, PD,
     1          YR(16), MO(16), DA(16), WK
C
C  Accesses crime database (tabular form) and user prediction
C  monitoring file
C
      CHARACTER*1 DUMMY
      CHARACTER*8 PERIOD
      CHARACTER*10 DATETX
C
C  Machine and user predictions:  Means and Variances
C
      REAL*4 UPRED(32), LWK(32), MARRAY(32), MPRED(32), UVAR(32)
      REAL*4 MVAR(32), PRED(32)
      INCLUDE 'A:SYSREG.FOR'
C
C  Attach beatwise tabular database for past 16 weeks
C
      OPEN (3,FILE='TABCRM',FORM='UNFORMATTED')
      DO 50 I = 1, 16
C
C  Week ending for each weekly record
C
        READ (3) YR(I), MO(I), DA(I)
C
C  Beatwise crime array for each record
C
   50   READ (3) (CRIMES(I,J), J = 1, 32)
      CLOSE (3)
C
C  Machine makes its prediction
C
      DO 98  I = 1, 32
        MARRAY(I) = FLOAT(CRIMES(1,I))
   98   LWK(I)    = FLOAT(CRIMES(2,I))
      CALL MPPR (MARRAY, LWK, MPRED, MVAR, 32)
C
C  Initial map is of machine prediction
C
      WK = 1
      PERIOD = ' 7 Days '
C
C  Week ending of next week
C
      CALL DT2NM(YR(1), MO(1), DA(1), NDAT)
      NDAT = NDAT + 7
      CALL NM2DT(PY, PM, PD, NDAT)
      WRITE (DATETX, '(I2,1H/,I2,1H/,I4)') PD, PM, PY
C
C Beginning of iterative user modification loop
C
   66 CALL MODE(16)
      CALL BOXES(15)
```

```
      CALL PUTTXT(20,1,'South Gosforth Subdivision')
      CALL PUTTXT(22,2,'Household Burglaries')
      CALL CHKEY
      CALL PUTTXT(58, 2,'     Menu :-')
      CALL PUTTXT(58, 4,'<T> = Tables      ')
      CALL PUTTXT(58, 6, <V> = Variability ')
      CALL PUTTXT(58, 8, <A> = Association ')
      CALL PUTTXT(58,10,'<E> = Exit to Menu')
      CALL PUTTXT(19,3,PERIOD//'Ending '//DATETX)
   51 CONTINUE
C
C  Shade in the beats
C
      DO 100 I = 1, 32
        CVALS(I) = 11
        IPVAL    = INT(PRED(I) + 0.5)
        PVALS(I) = 1
        IF (IPVAL .LT. 8) PVALS(I) = 2
        IF (IPVAL .LT. 6) PVALS(I) = 4
        IF (IPVAL .LT. 4) PVALS(I) = 8
        IF (IPVAL .LT. 2) PVALS(I) = 0
  100   CONTINUE
      CALL MAP('BEATS',PVALS,CVALS,32)
      DO 110 I = 1, 32
        CVALS(I) = 15
        PVALS(I) = 1
  110   CONTINUE
      CALL MAP('BORDERS',PVALS,CVALS,32)
C
C  Begin the Main Menu Loop for prediction
C
  500 CALL GETKEY(ICHCE)
C
C  Find associated beats (if key A is pressed)
C
      IF (ICHCE .EQ. 65) THEN
        CALL ASSOC
      END IF
C
C  Find beats with most variance (if Key V is pressed)
C
      IF (ICHCE .EQ. 86) THEN
        CALL VARBT(MVAR, UVAR, CVALS, 32)
        CALL MAP('BEATS',PVALS,CVALS,32)
      END IF
C
C  Put tables up ( if key T is pressed)
C
      IF (ICHCE .EQ. 84) THEN
        DO 600 I = 1, 32
          UPRED(I) = 0.375 + MPRED(I)
  600     UVAR(I)  = SQRT(UPRED(I))
C
C  Draw tables of predictions
C
        CALL TABLES(UPRED,UVAR,32)
C
C  Initiate past user prior performance assessment routine
C
        CALL USRCAL(UPRED,UVAR)
C
```

```
C  Merge the combination
C
        CALL MERGE (UPRED,UVAR,MPRED,MVAR,PRED)
        GO TO 66
      END IF
C
C  If E not pressed,  return to menu; else exit menu loop
C
      IF (ICHCE .NE. 69) GO TO 500
      CALL MODE(3)
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE CHKEY
C
C Prints the key for the prediction map
C
      CALL KEYBOX(45, 5,'0 -< 2',11,0)
      CALL KEYBOX(45, 7,'2 -< 4',11,8)
      CALL KEYBOX(45, 9,'4 -< 6',11,4)
      CALL KEYBOX(45,11,'6 -< 8',11,2)
      CALL KEYBOX(45,13,'8 -<  ',11,1)
      CALL PUTTXT(40,17,'   Forecasted  ')
      CALL PUTTXT(40,18,'Crimes per Week')
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE MPPR(CRIMES, LWK,  PRED, VAR, N)
C
C
C  Machine Prediction PRocedure (Hence MPPR)
C
C
C
      REAL*4 CRIMES(N), MEANS(50), DIST(50,50), ALPHA, NEWMN, CMETRC
      REAL*4 HHOLDS(50), UPPER(N), VRNCE(50), ACORR, LWK(32), PRED(N)
      REAL*4 VAR(N), BTMEAN(32), SHAPE(33), VFAC
      INTEGER ADJLST(9,50), SIDES, BTCOUN(32)
      DATA VFAC /0.007/
C
C  Read in the adjacency lists,  the inverse distances and the data ...
C
      OPEN (11,FILE='DISTS')
      OPEN (14,FILE='ADJLST')
      OPEN (15,FILE='HHOLDS', FORM='UNFORMATTED')
      DO 110 I = 1, N
        READ (14,100) ADJLST(1,I), (ADJLST(J,I),J=2,ADJLST(1,I) + 1)
  100   FORMAT (50I2)
  110 CONTINUE
      READ (15)  (HHOLDS(I), I = 1, N)
C
C  Compute transformed means and distances (after Bartlett, 1948)
C  and compensation for household densities
C
      DO 120 I = 1, N
        VRNCE(I) = 1.0 / (4.0*HHOLDS(I))
  120 MEANS(I) = SQRT(CRIMES(I)/HHOLDS(I))
```

```
      DO 130 I = 1, N
  130 READ (11,*) (DIST(J,I),J=1,N)
      CLOSE (14)
      CLOSE (15)
      CLOSE (11)
C
C  Get the Space-Time Autoregression characteristics
C
      OPEN (14, FILE='STAR')
      READ (14,* ) SHAPE
      CLOSE (14)
C
C  Now perform the prediction : read in the mean level estimates
C  and the counts of how many observations they are based on
C
      OPEN (15, FILE='BTMEAN', FORM='UNFORMATTED')
      READ (15) BTMEAN
      READ (15) BTCOUN
      CLOSE (15)
      DO 160 I = 1, 32
        PRED(I) = (MEANS(I) - BTMEAN(I))*SHAPE(33)
        DO 150 J = 1, ADJLST(1, I)
          K = ADJLST(J+1,I)
  150     PRED(I) = PRED(I) + (MEANS(K) - BTMEAN(K))*SHAPE(I)/DIST(I,K)
        PRED(I) = PRED(I) + BTMEAN(I)
        VAR(I)  = (1.0 + 1.0/BTCOUN(I)) * VFAC
  160   CONTINUE
C
C  Save results for calibration
C
      OPEN (20, FILE= 'COMP.PRD', FORM='UNFORMATTED')
      WRITE (20, PRED)
      WRITE (20, VAR)
C
C  Transform the estimates back to numbers of crimes
C
      DO 190 I = 1, N
        PRED(I) = (PRED(I))**2 * HHOLDS(I)
  190   VAR(I)  = VAR(I) ** 2 * HHOLDS(I) * 2.0
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE TABLES(PRED,UVAR,N)
C
C Prints tables of past prediction:  Also controls user prediction
C
      REAL*4 PRED(N), UVAR(N)
      CHARACTER*4 BEAT(50)
      CHARACTER*5 NEWVAL
      CHARACTER*5 YEL, CYAN
      CHARACTER*8 REDONW, WHOB
      CHARACTER*52 STEXT
      INTEGER*1 ESC
C
C  Screen control codes
C
      DATA ESC/27/
      WRITE (YEL   ,'(A1,A)') ESC,'[33m'
      WRITE (CYAN  ,'(A1,A)') ESC,'[36m'
```

```
      WRITE (REDONW,'(A1,A)') ESC,'[31;47m'
      WRITE (WHOB  ,'(A1,A)') ESC,'[37;40m'
C
C  Read beat names
C
      OPEN (1,FILE='BEATS',FORM='UNFORMATTED')
      DO 100 I = 1, N
  100   READ (1) BEAT(I)
      CLOSE (1)
C
C  Clear screen and print table
C
      CALL MODE (3)
      CALL PUTTXT(23,2,'Predicted Burglaries for Next Week')
      DO 110 I = 1,8
        WRITE (STEXT,'(4(A,2X,F5.1,2H |))')
     1  (BEAT(I+J),PRED(I+J), J = 0, 24, 8)
        CALL PUTTXT(14,7+I,YEL//STEXT//WHOB)
 110    CONTINUE
      CALL PUTTXT(14, 17, CYAN//'<R> = Return to map')
      CALL PUTTXT(14, 19, '<M> = Modify Forecast'//WHOB)
C
C  Menu loop for prediction adjustment
C
 150  CALL GETKEY(ICHCE)
C
C  Modify the machine predictions (ie adjust user prior if M pressed)
C
      IF (ICHCE .EQ. 77) THEN
C
C  Set up menu for beat modification
C
        CALL PUTTXT(15, 20, 'Use <Z> and <X> to point at beats')
        CALL PUTTXT(15, 21, 'Use <R> to return to map')
        CALL PUTTXT(15, 22, 'Use <B> to select a beat to modify')
        IBPTR = 1
        IXPTR = 13
        IYPTR = 8
 120    CALL PUTTXT(IXPTR, IYPTR, ' ')
        IXPTR = ((IBPTR - 1) /8)*13 + 18
        IYPTR = MOD(IBPTR - 1, 8) + 8
        CALL PUTTXT(IXPTR, IYPTR, REDONW//'<'//WHOB)
C
C  Menu for Beat Modification
C
        CALL GETKEY(ICHCE2)
C
C  Indicate last  beat if Z pressed
C
        IF (ICHCE2.EQ. 90) THEN
          IBPTR = IBPTR - 1
          IF (IBPTR .EQ. 0) IBPTR = 32
          GO TO 120
        END IF
C
C  Indicate next beat if X pressed
C
        IF (ICHCE2.EQ. 88) THEN
          IBPTR = IBPTR + 1
          IF (IBPTR .EQ. 33) IBPTR = 1
          GO TO 120
```

```
      END IF
C
C  Select indicated beat for modification if B pressed
C
      IF (ICHCE2.EQ. 66) THEN
C
C  New submenu loop :  adjust prediction level
C
        CALL PUTTXT(03,23,'Make forecast 1) Much larger ')
        CALL PUTTXT(43,23,'2) Slightly larger')
        CALL PUTTXT(03,24,'              3) Slightly less')
        CALL PUTTXT(43,24,'4) Much less')
        CALL PUTTXT(59,24,'5) Correct')
 141    CALL GETKEY(ICHCE3)
        ICHCE3 = ICHCE3 - 48
C
C  Numeric case statement
C
        GOTO (210, 220, 230, 240, 242), ICHCE3
        GO TO 141
 210    PRED(IBPTR) = PRED(IBPTR)*2.0 + 1.0
        GO TO 140
 220    PRED(IBPTR) = PRED(IBPTR)*1.25
        GO TO 140
 230    PRED(IBPTR) = PRED(IBPTR)/1.25
        GO TO 140
 240    PRED(IBPTR) = (PRED(IBPTR) - 1.0)/2.0
        IF (PRED(IBPTR) .LT. 0.0) PRED(IBPTR) = 0.0
        GO TO 140
 140    WRITE (NEWVAL,'(F5.1)') PRED(IBPTR)
        CALL PUTTXT(IXPTR+2, IYPTR, REDONW//NEWVAL//WHOB)
        GO TO 141
C
C  Clean up after previous menu
C
 242    CALL PUTTXT(03,23,'                              ')
        CALL PUTTXT(43,23,'                  ')
        CALL PUTTXT(03,24,'                               ')
        CALL PUTTXT(43,24,'            ')
        CALL PUTTXT(59,24,'          ')
C
C  Now similar menu to obtain variance of user prior
C
        CALL PUTTXT(3,23,'    How certain is the prediction ?     ')
        CALL PUTTXT(45,23,'  1) Very Certain                ')
        CALL PUTTXT(3,24,'   2) Within usual variability          ')
        CALL PUTTXT(45,24,'  3) Likely to vary a lot        ')
 244    CALL GETKEY(ICHCE4)
C
C  Direct conversion to an integer
C
        ICHCE4 = ICHCE4 - 48
        IF (ICHCE4 .LT. 1 .OR. ICHCE4 .GT. 3) GO TO 244
        IF (ICHCE4 .EQ. 1) UVAR(IBPTR) = SQRT(PRED(IBPTR))/2.0
        IF (ICHCE4 .EQ. 2) UVAR(IBPTR) = SQRT(PRED(IBPTR))
        IF (ICHCE4 .EQ. 3) UVAR(IBPTR) = SQRT(PRED(IBPTR))*1.5
        CALL PUTTXT(5,23,'                                          ')
        CALL PUTTXT(45,23,'                                  ')
        CALL PUTTXT(5,24,'                                          ')
        CALL PUTTXT(45,24,'                                  ')
        GO TO 120
```

```
        END IF
C
C  Set up an exit if R (for Return) is pressed
C
        IF (ICHCE2 .EQ. 82) ICHCE = 82
        IF (ICHCE2 .NE. 82) GO TO 120
      END IF
C
C  Set up an exit from main user prior menu
C
      IF (ICHCE .NE. 82) GO TO 150
C
C  Save the predictions for future assessment
C
      OPEN(17, FILE='USER.PRD', FORM='UNFORMATTED')
      WRITE (17) SPMEAN, SPDEV
      CLOSE (17)
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE USRCAL(SPMEAN,SPDEV)
C
C  User calibration routine
C  Uses a Gaussian integration technique and interpolation to
C  evaluate the performance function convolution (which adjusts
C  the user prior
C
      INTEGER BEAT
      REAL*4 F(0:40,32), IGRAL, K, X, XL, INC, FMODIF(10)
      REAL*4 SPMEAN(32), SPDEV(32), ACTUAL, NORMAL, IGRAL2, IGRAL0
      REAL*4 ABSCIS(10), W(10), SQR2, ROOTPI, HHOLDS(32)
C
C   Gaussian 10-point rule constants
C
       DATA ABSCIS/
     +           -3.436158289955701,
     +           -2.532731063278602,
     +           -1.756683225546651,
     +           -1.036610579734708,
     +          -0.3429012445078736,
     +           0.3429012445078770,
     +            1.036610579734711,
     +            1.756683225546650,
     +            2.532731063278606,
     +            3.436158289955692  /
       DATA W/
     +          0.7640431012181091E-05,
     +          0.1343645422662423E-02,
     +          0.3387438628418394E-01,
     +          0.2401385531552589    ,
     +          0.6108624863809554    ,
     +          0.6108624863809539    ,
     +          0.2401385531552572    ,
     +          0.3387438628418407E-01,
     +          0.1343645422662396E-02,
     +          0.7640431012181565E-05 /
       DATA  SQR2 /1.414213562/
       DATA  ROOTPI/0.564189584/
C
```

```
C Obtain performance function of cumulants
C
      OPEN(14, FILE='USER.PER', FORM='UNFORMATTED')
      DO 100 BEAT = 1, 32
        READ (14) (F(J,BEAT), J = 0, 40)
  100   CONTINUE
      CLOSE (14)
C
C  Data to transform between crime counts and normalised data
C
      OPEN(15, FILE='HHOLDS', FORM='UNFORMATTED')
      READ (15) HHOLDS
      CLOSE (15)
      DO 105 BEAT = 1, 32
        SPDEV(BEAT)  = SPDEV(BEAT)/SQRT(SPMEAN(BEAT))
        SPDEV(BEAT)  = SPDEV(BEAT)/SQRT(HHOLDS(BEAT))
  105   SPMEAN(BEAT) = SQRT(SPMEAN(BEAT)/HHOLDS(BEAT))
      DO 900 BEAT = 1, 32
        IGRAL0 = 0.0
        IGRAL  = 0.0
        IGRAL2 = 0.0
        DO 200 I = 1, 10
C
C Interpolate the performance function of the cumulant
C
          XL = NORMAL(ABSCIS(I)*SQR2)
          NEARPT = INT(40.0*XL)
          INC = XL - FLOAT(NEARPT)*0.025
          XL = F(NEARPT,BEAT) + INC*(F(NEARPT+1,BEAT)-F(NEARPT,BEAT))
          FMODIF(I) = XL
C
C  Estimate the Zeroth, First and Second moments of the modified
C  Prior  using  Gauss-Hermite  approximation  for integral  of
C  f(x)*exp(-x**2) over the real line.
C
C
          IGRAL0 = IGRAL0 + FMODIF(I)*W(I)
          IGRAL  = IGRAL +
     1     FMODIF(I)*W(I)*(SPMEAN(BEAT) + SPDEV(BEAT)*SQR2*ABSCIS(I))
          IGRAL2 = IGRAL2 +
     1     FMODIF(I)*W(I)*(SPMEAN(BEAT) + SPDEV(BEAT)*SQR2*ABSCIS(I))**2
  200     CONTINUE
C
C  From these, deduce the mean and standard deviations of the modified
C  distributions....
C
        IGRAL = IGRAL/IGRAL0
        IGRAL2 = IGRAL2/IGRAL0 - IGRAL**2
        SPMEAN(BEAT) = IGRAL
        SPDEV(BEAT)  = SQRT(IGRAL2)
  900   CONTINUE
        DO 910 BEAT = 1, 32
          SPMEAN(BEAT) = SPMEAN(BEAT)**2*HHOLDS(BEAT)
          SPDEV(BEAT)  = 2.0*SPDEV(BEAT)**2*HHOLDS(BEAT)
  910     CONTINUE
      RETURN
      END
C
C*********************************************************************
C
      FUNCTION NORMAL(Z)
```

```
      REAL*4 X, T, B(5), P, F, SQ2PI, NORMAL, Z
      LOGICAL LOWER
C
C  Hastings approximation for area under the normal curve
C
C   Error < 1.5E-7
C
      DATA B /0.254829592,
     1       -0.284496736,
     2        1.421413741,
     3       -1.453152027,
     4        1.061405429/
      DATA P /0.23164189/
      X = Z
      LOWER = (X .LT. 0.0)
      IF (LOWER) X = -X
      T =  1.0 / (1.0 + P*X)
      F = B(5)
      DO 100 I = 2, 5
  100   F = F*T + B(6-I)
      VALUE  = 0.5*T*F*EXP(-X**2/2.0)
      IF (LOWER) THEN
        NORMAL = VALUE
      ELSE
        NORMAL = 1.0 - VALUE
      END IF
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE MERGE(UPRED, UVAR, MPRED, MVAR, PRED)
C
C  Combine user and machine priors
C
      REAL*4 UPRED(32), UVAR(32), MPRED(32), MVAR(32), PRED(32)
      INTEGER BEAT
C
C  Weighted mean merge (assummes no correlation for simulataneous
C  future events --- see text in Chapter 5.
C
      DO 100 BEAT = 1, 32
        PRED(BEAT) = MPRED(BEAT)*UVAR(BEAT) + UPRED(BEAT)*MVAR(BEAT)
        PRED(BEAT) = PRED(BEAT)/(UVAR(BEAT) + MVAR(BEAT))
  100 CONTINUE
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE ASSOC
C
C  Joins most associated beats together with lines
C  Currently does this entirely on fixed Space-Time Autoregression
C  model ....
C
      INTEGER ADJLST(9,50)
      REAL SHAPE(33), CENX(32), CENY(32), DIST(32,32)
      OPEN (14, ADJLST)
C
C  Get adjacencies
```

```
C
      DO 110 I = 1, N
        READ (14,100) ADJLST(1,I), (ADJLST(J,I),J=2,ADJLST(1,I) + 1)
  100   FORMAT (50I2)
  110 CONTINUE
C
C  Get space-time autoregression model
C
      OPEN (13, FILE='STAR')
      READ (13,* ) SHAPE
      CLOSE (13)
C
C  Get distances
C
      OPEN (11, FILE='HHOLDS')
      DO 130 I = 1, 32
  130 READ (11,*) (DIST(J,I),J=1,32)
C
C  Get centroids
C
      OPEN (10, FILE='CENTS',FORM='UNFORMATTED')
      READ (10) CENX
      READ (10) CENY
      CLOSE(10)
C
C  Scan for sufficient association
C
      DO 160 I = 1, 32
        DO 150 J = 1, ADJLST(1, I)
          K = ADJLST(J+1,I)
C
C  Join associated zones with a line
C
          IF (SHAPE(K)*SHAPE(I)/DIST(I,K) .GT. 0.25) THEN
  150       CALL LINE(CENX(I), CENY(I), CENX(K), CENY(K))
  160   CONTINUE
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE VARBT(UVAR, MVAR, CVALS, NZONES)
C
C   Subroutine to highlight most variable beats,  in terms
C   of predictor distribution
C
      REAL*4 UVAR(NZONES), MVAR(NZONES), CVALS(NZONES)
      REAL*4 OVAR(50), SOVAR
C
C   Combine variances
C
      DO 100 I = 1, NZONES
  100   OVAR(I) = 1.0/(1.0/MVAR(I) + 1.0/UVAR(I))
C
C    Highlight the top quarter
C
      CALL SORT(OVAR, SOVAR,  NZONES)
      DO 110 I = 1, NZONES
        IF (OVAR(I) .GT. SOVAR(24)) CVALS(I) = 13
  110 CONTINUE
      RETURN
```

```
END
```

```
      PROGRAM MONTOR
C
C     ****** Listing 6.13 *****
C
C   Monitor the performance of the running mean estimator in the
C   predictor and set it to re-estimate if continuous bad performance
C   is observed --- also calibrate the user performance function
C
      INTEGER*4 FLAG(32), BTCOUN(32), COUNT(32), HH(32), D1, D2, D3
      REAL      BTMEAN(32), MPRED(32), MVAR(32), HHF(32)
      REAL      UPRED(32), UVAR(32)
      REAL*4 F(0:40,32), IGRAL, K, X, XL, INC, FMODIF(10)
      REAL*4 NORMAL, IGRAL2, IGRAL0, ALPHA
      REAL*4 ABSCIS(10), W(10), SQR2
      INTEGER BEAT
      DATA  ALPHA /0.6/
C
C   Read in the exception monitor (see Chapter 5)
C
      OPEN (1, FILE='COMP.MON', FORM='UNFORMATTED')
      READ (1) FLAG
      CLOSE (1)
C
C   Read in the machines predictions
C
      OPEN (2, FILE='COMP.PRD',FORM='UNFORMATTED')
      READ (2) MPRED
      READ (2) MVAR
      CLOSE (2)
C
C   Read in the current mean estimates
C
      OPEN (3, FILE='BTMEAN', FORM='UNFORMATTED')
      READ (3) BTMEAN, BTCOUN
      CLOSE (3)
C
C   Read in the household counts
C
      OPEN (4, FILE='HHOLDS', FORM='UNFORMATTED')
      READ (4) HH
      DO 100 I = 1, 32
  100   HHF(I) = FLOAT(HH(I))
      CLOSE (4)
C
C   Read in the actual figures
C
      OPEN (7, FILE='TABCRM', FORM='UNFORMATTED')
      READ (7), D1, D2, D3
      READ (7)  COUNT
      CLOSE (7)
C
C Read in user predictions
C
      OPEN (8, FILE = 'USER.PRD',  FORM='UNFORMATTED')
      READ (8) UPRED
      READ (8) UVAR
      CLOSE(8)
C
C Read in performance function for user
C
      OPEN (9, FILE='USER.PER', FORM='UNFORMATTED')
```

```
      DO 160 BEAT = 1, 32
  160   WRITE (9) (F(I,BEAT), I = 0, 40)
      CLOSE(9)
C
C Monitor for outstanding values of machine predictions
C
      DO 110 I = 1, 32
        IF (SQRT(COUNT(I)/HH(I)) .GT. MPRED(I) + 1.96*MVAR(I)) THEN
C
C Outlier : increment warning flag and update estimator
C if required
C
          FLAG(I) = FLAG(I) + 1
          IF (FLAG(I) .EQ. 2) THEN
            BTMEAN(I) = SQRT(COUNT(I)/HH(I))
            BTCOUN(I) = 1
          END IF
        ELSE
C
C   Normal observation : increment posterior mean (see Chapter 5)
C
          FLAG(I) = 0
          BTMEAN(I) = BTMEAN(I) * BTCOUN(I) + COUNT(I)
          BTCOUN(I) = BTCOUN(I) + 1
          BTMEAN(I) = BTMEAN(I) / BTCOUN(I)
        END IF
  110 CONTINUE
      DO 115 BEAT = 1, 32
C
C Numerical Implimentation of the Morris calibration
C
       SPMHH = UPRED(BEAT) / HH(BEAT)
       SPHH  = SQRT(UVAR(BEAT)))/(2.0*HH(BEAT))
       ACTHH = SQRT((COUNT(BEAT) + 0.375)/ HH(BEAT))
       PROB  = NORMAL((ACTHH  - SPMHH )/SPHH )
C
C Beta(A,1)  distribution for PROB --- update estimator
C
       DO 120 I = 0, 40
         X = FLOAT(I) / 40.0
C
C  Avoid division by zero!
C
          IF (PROB .LT. 0.2) PROB=0.2
          IF (PROB.GT.0.8) PROB=0.8
          A = PROB/(1-PROB)
  120     F(I,BEAT) = F(I,BEAT)+(X**A*(1-X))**ALPHA
C
C Make it integrate to unity
C
       IGRAL = 0.0
       DO 130 I = 1, 39
  130    IGRAL = IGRAL + F(I,BEAT)
       IGRAL = IGRAL*2.0 + F(0,BEAT) + F(40,BEAT)
       IGRAL = IGRAL * 0.0125
       DO 140 I = 0, 40
  140    F(I,BEAT) = F(I,BEAT) / IGRAL
C
C Get next probability
C
  115  CONTINUE
```

```
C
C   Write out the exception monitor (see Chapter 5)
C
      OPEN (1, FILE='COMP.MON')
      WRITE (1) FLAG
      CLOSE (1)
C
C   Write out the running mean estimates
C
      OPEN (3, FILE='BTMEAN')
      WRITE (3) BTMEAN
      WRITE (3) BTCOUN
      CLOSE (3)
C
C   Write out the user performance function
C
      OPEN (9, FILE='USER.PER', FORM='UNFORMATTED')
      DO 150 BEAT = 1, 32
  150   WRITE (9) (F(I,BEAT), I = 0, 40)
      CLOSE(9)
      STOP
      END









      STOP
      END
      FUNCTION NORMAL(Z)
      REAL*4 X, T, B(5), P, F, SQ2PI, NORMAL, Z
      LOGICAL LOWER
C
C  Hastings approximation for area under the normal curve
C
C   Error < 1.5E-7
C
      DATA B /0.254829592,
     1       -0.284496736,
     2        1.421413741,
     3       -1.453152027,
     4        1.061405429/
      DATA P /0.23164189/
      X = Z
      LOWER = (X .LT. 0.0)
      IF (LOWER) X = -X
      T =  1.0 / (1.0 + P*X)
      F = B(5)
      DO 100 I = 2, 5
  100   F = F*T + B(6-I)
      VALUE  = 0.5*T*F*EXP(-X**2/2.0)
      IF (LOWER) THEN
        NORMAL = VALUE
      ELSE
        NORMAL = 1.0 - VALUE
      END IF
      RETURN
      END
```

C H A P T E R 7

THE USERS VIEWPOINT

7.1 Introduction

It is surprising that although great advances have recently been made in computerised cartographic and geographical information systems, as yet little attention has been focused on the man-machine interface of such software. However, the maps produced are basically a means of communicating information, there is a message, and if the correct impressions are to be given then careful thought must be given to providing acceptable map display formats. In addition to mapping aspects, consideration of ease of use of other aspects of software designed in this study is necessary. Clearly, a system such as this, in its working environment will be used frequently. Difficulties in operation may lead to erroneous data being entered into the system, and may discourage potential users from accessing the information (in terms of predictions, map patterns, and so on) that the system has to offer.

This is particularly important in the case of Bayesian systems such as this. Since predictions of crime rates are based at least partly on input from police users, it is important that they have a fluent dialogue with the machine. Badly presented data or control options will affect the operators understanding of the system, which will in turn alter the predictions obtained.

Thus, in this chapter, investigation into police user interaction with the crime pattern analysis system is to be carried out. This is proposed on two levels. Firstly, a study of map visualisation will be carried out, using a sample of all of the police officers in a subdivision. This will take the form of a questionnaire survey, the subjects being shown several different map formats and asked to objectively evaluate them. The purpose of this is to gain a general overview of the map formats that are preferred, and are thought to convey relevant information in an easily assimilated manner. Secondly, an individual user will be allowed to operate the prototype system, and enter data corresponding to crimes occurring over a two-month period. After this, the user will be interviewed, and comments about the "look and feel" of the system will be considered. Clearly, this will be a more subjective evaluation of the system.

The second study should serve two purposes. Firstly, it is impossible to design a system "from the drawingboard" to be without errors. Certain problems and limitations may not occur to the designer, but will only become apparent when the system is put into use. A trial usage of this kind should identify some of the major omissions or design flaws before the system is installed in a "live" environment. Viewed in another way, it allows a second person to comment on system design, and engage in a dialogue with the initial designer. The second purpose that this trial serves is to allow the performance of the system to be assessed in a working environment. Different features of the system can be compared for effectiveness when real crime data is input to the system.

The aim of the chapter may be summarised as an evaluation of the interaction of human users with geographical analysis software, and in particular with the system developed in this PhD.

7.2 A Map Visualisation Study

The objective of this study is firstly to identify a set of possible map formats that may be used to convey information about crime rates, and having done this carry out a survey of responses of police officers to these different formats. The survey is to be carried out at a subdivision of the Northumbria Police, using data of archived household burglary reports occurring within that subdivision.

7.2.1 Possible Map Formats

Before designing a survey on response to mapped representation of crime data, it is necessary to outline the set of options for such map displays. First, the general types of map which may be used will be considered. These may be split into three main categories.

i) Maps based on point data representation.

ii) Maps based on data aggregated to areas.

iii) Isopleth maps for estimated density of occurrence, based on point data.

In all cases, the data being referred to are the grid references to the incidences of reported household burglaries over a given period of time. In format i) these grid references are plotted directly onto a map of the subdivision, giving a pin-map format, as in figure 7.2. Alternatively, the point data can be analysed to give an estimated 'density surface' over the region of study, and contour maps in the format of type iii) can be compiled as in figure 7.1. At the subdivisional headquarters where this survey is based, data for incidence counts aggregated over foot beat areas is kept on a weekly basis. This data may be used to produce maps of type ii). In addition to this choice, the amount of boundary information shown on maps could be varied. In addition to showing the outline of the entire subdivision, foot beat boundaries may also be added. If they are, then beat-related decision making and forecasting may be aided, but a geographical analysis based on other areal units may be confused by the inclusion of this extra visual information.

Similarly, there is the question of whether text labels for place names, and local geographical features should be included on the maps. In favour of this, police officers may find that maps are easier to interpret in terms of the spatial relation between crime occurrence and named local areas and landmarks, rather than in terms of the more abstract notions of beat boundaries and subdivisional borders. Against, as in the previous point, inclusion of further information on maps can lead to confused, cluttered displays. It was

eventually decided, after informal conversations with senior police officers, that the number of maps required to show each format with varying degrees of extra label information would be inoperably large, so that maps with a fixed amount of label information would be used. It is hoped that a desire for more or less detial of this type will then be picked up in the 'comment' section of the questionaires.

7.2.2 Method Of Map Production

The maps that will be used in this survey will be produced using the GIMMS package, excepting the contour maps. GIMMS can produce point-pattern and areally aggregated maps, with or without beat boundaries, and also offers the option of text labelling on the maps it draws. The formats of maps that GIMMS may produce will now be considered in greater detail:-

i) Spatially Aggregated Maps

GIMMS offers several options for map display of spatially aggregated data. These are accessed via the *MAP command. The main types under consideration are LABEL, POINT, and AREA. In a LABELed map, each beat region is annotated by the actual value associated with it in the data file. The size of the text used to write this value may either vary in proportion to the value, or be fixed. This is controlled by the *SYMBOLISM command, issued earlier in the GIMMS control deck. In a POINT map, a symbol is drawn at the centroid of

each beat. The size of this symbol varies in proportion to the number of crimes aggregated to each beat. The type of symbol could be a square, a circle or other options, and may be shaded in various ways. As before, all of these factors are controlled using the *SYMBOLISM command. Note that it is distances on the symbols, and not areas, that vary in proportion to the crime counts.

Finally, an AREA map is simply a choropleth map, where the shading of each beat shows into which category of crime incidence count it falls. The shading styles for different classes are selected by *SYMBOLISM once more.

ii) Point Pattern Maps

In addition to the above, GIMMS also offers an option for plotting point patterns onto maps, as follows. Grid references are stored in a POINT type file, which may then be drawn using the drawmap command with the CROSS option. This may be overdrawn onto a file containing the subdivisional outline and beat boundaries, of type AREA or SEGMENT, providing a crime incidence map over a given time period.

As well as the above formats offered by GIMMS, some form of contour mapping will also be required. The package SURFACE2 allows contour maps to be produced reasonably simply, and so will be used here. Unfortunately this package has no facilities for text labelling on the maps it produces, but on the operating system

that the software is implemented plotters are not accessed immediately, but driven by a control file produced by the software, and this may be modified to include text after the contouring job is completed.

Given a set of coordinates for household burglary incidences, SURFACE2 can build up a set of contours for estimated 'household burglary density' over the subdivisional area, using two dimensional extrapolation methods. It should be borne in mind that such an extrapolation fits a model to an infinite number of points (a surface) from a finite sample, which could lead to strange sets of contours on occasion, particularly if the set of sample points is small or locally sparse. Thus, contour representation is probably most useful in regions of high crime incidence, and spurious contours in areas of low risk should be treated with caution.

Bearing this danger in mind, a better way of mapping density-based interpretations of the data would be to only plot contours of a single value, and draw these as boundaries to 'high-risk' areas. This avoids the pitfall of misinterpreting contours occurring in regions of low crimes. Both of the options, complete contouring and 'high risk area' indication, will be considered.

Enough options have now been discussed to decide which maps are to be included in the survey. Discussions between Police Inspectors at the subdivision chosen for the survey and myself

lead to some conclusions about the maps before the survey was designed. Firstly, it seems unlikely that coloured maps would be feasible on the target system, due to prohibitive costs for the equipment, so it would be difficult to justify their inclusion in this survey. Secondly, it was felt that beat boundaries and place names were essential on these maps, as they provide a geographical frame of reference for people using them. Therefore, the principal object of this survey is to investigate how the actual crime incidence data can best be overlaid on a map of the area of pre-specified format.

A total of seven map formats were chosen for the survey eventually. This set of maps covers all of the types of spatial data representation discussed above. Each of these is described below, and all seven are illustrated in figures 7.1-7.7.

Map Format A, Contour Map.

This map was produced using GIMMS and SURFACE2 output edited together.

Map Format B, Incidence Map.

This map was produced using grid references of household burglaries over a period of one week, plotting them on a subdivisional map using GIMMS, as described above.

Map Format C, Numbers of Crimes, aggregated by beat.

This was produced using MAPTYPE = LABEL in GIMMS.

Map Format D, Proportional Circles and Map Format E, Proportional Squares.

These were both produced using POINT symbolism in GIMMS, to show the same data as that in C, but using proportional symbols to label the beats.

Map Format F, High Risk Regions.

Part of the information from the contour map is re-digitised as a GIMMS area file, highlighting higher crime risk regions in the subdivision, as was discussed above.

Map Format G, Choropleth Map.

This is a simple choropleth map of reported burglaries in each beat during the week before the map was created. High crime rate is related with high risk, and darker shading represents higher crime risk in a beat than lighter shading.

Thus there is now a set of definitions for the seven maps that will be used in the survey. Although they were generated using output from the packages GIMMS and SURFACE2, similar maps could be created on a micro without recourse to these packages, and the results of this survey should point the way for future decisions for implementing graphics routines in the final crime prediction package. In addition, many micros will offer the option of colour graphics, which will enhance the display of information.

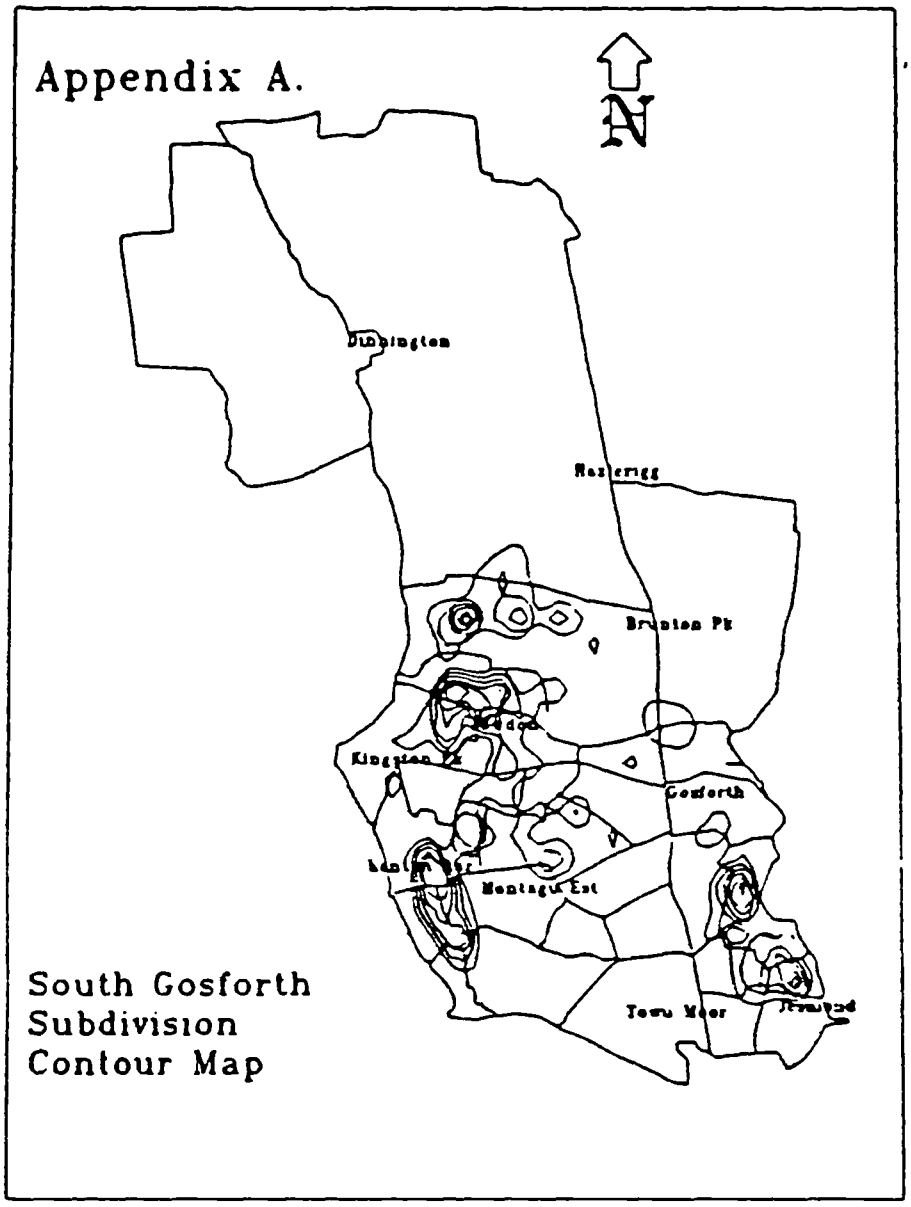

Appendix A.
N
Gosforth
South Gosforth
Subdivision
Contour Map

Figure 7.1

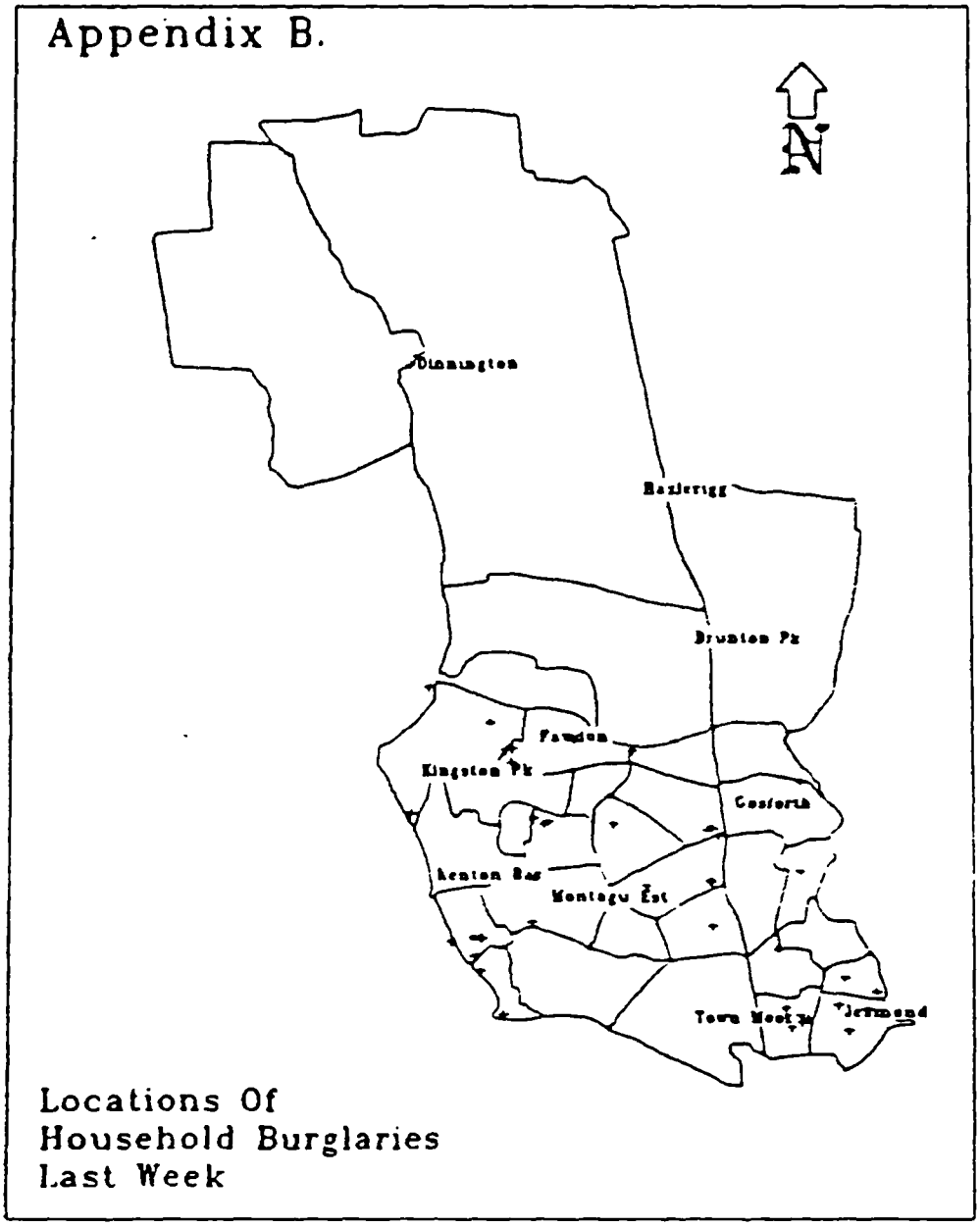

Appendix B.
N
Dinnington
Brunton Pk
Kingston Pk
Gosforth
Montagu Est
Locations Of
Household Burglaries
Last Week

Figure 7.2

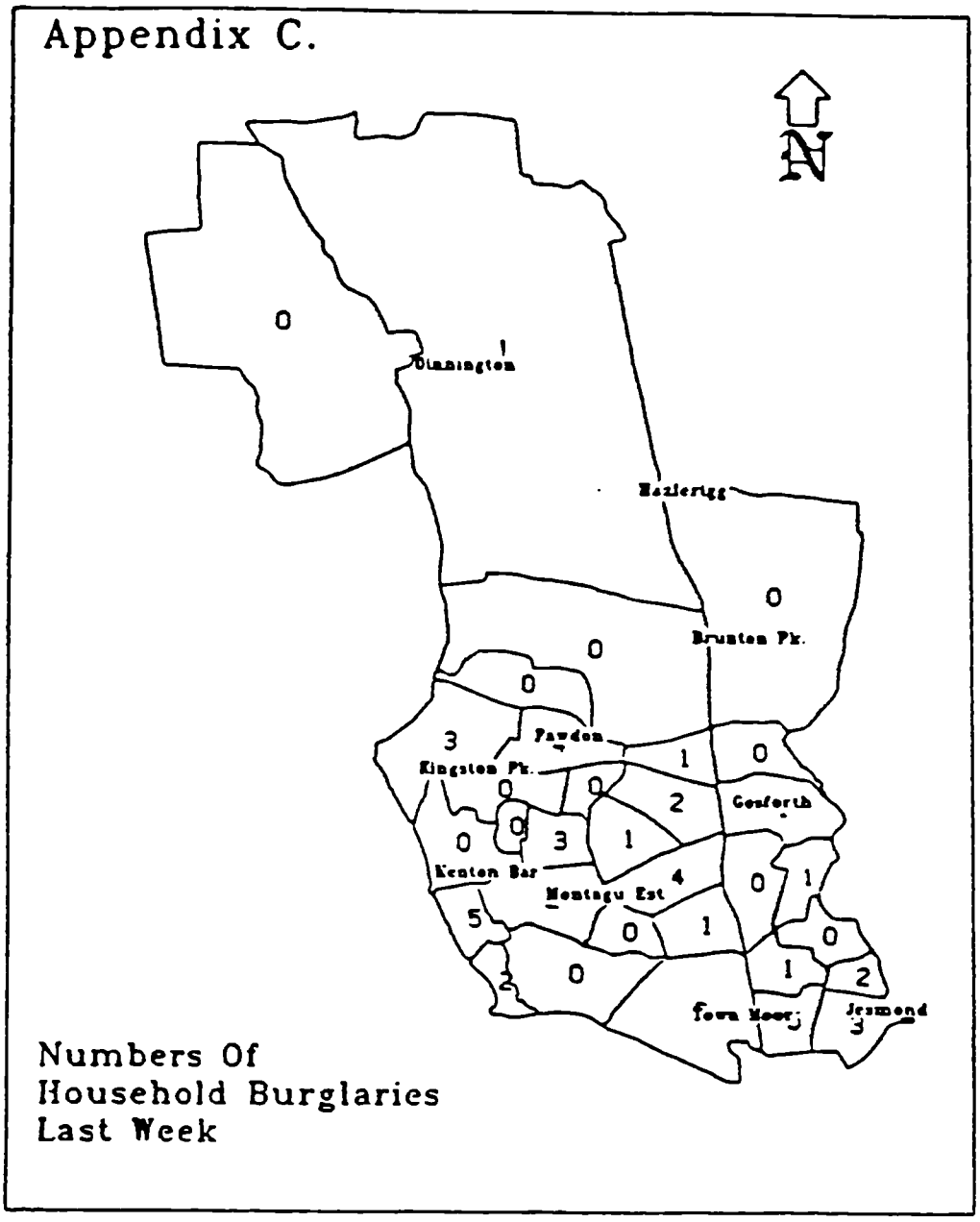

Appendix C.
N
Dunnington
Hazlerigg
Brunton Pk.
Pawdon
Kingston Pk.
Gosforth
Kenton Bar
Montagu Est
Fawn Moor
Jesmond
0
3
1
2
4
5
Numbers Of
Household Burglaries
Last Week

Figure 7.3

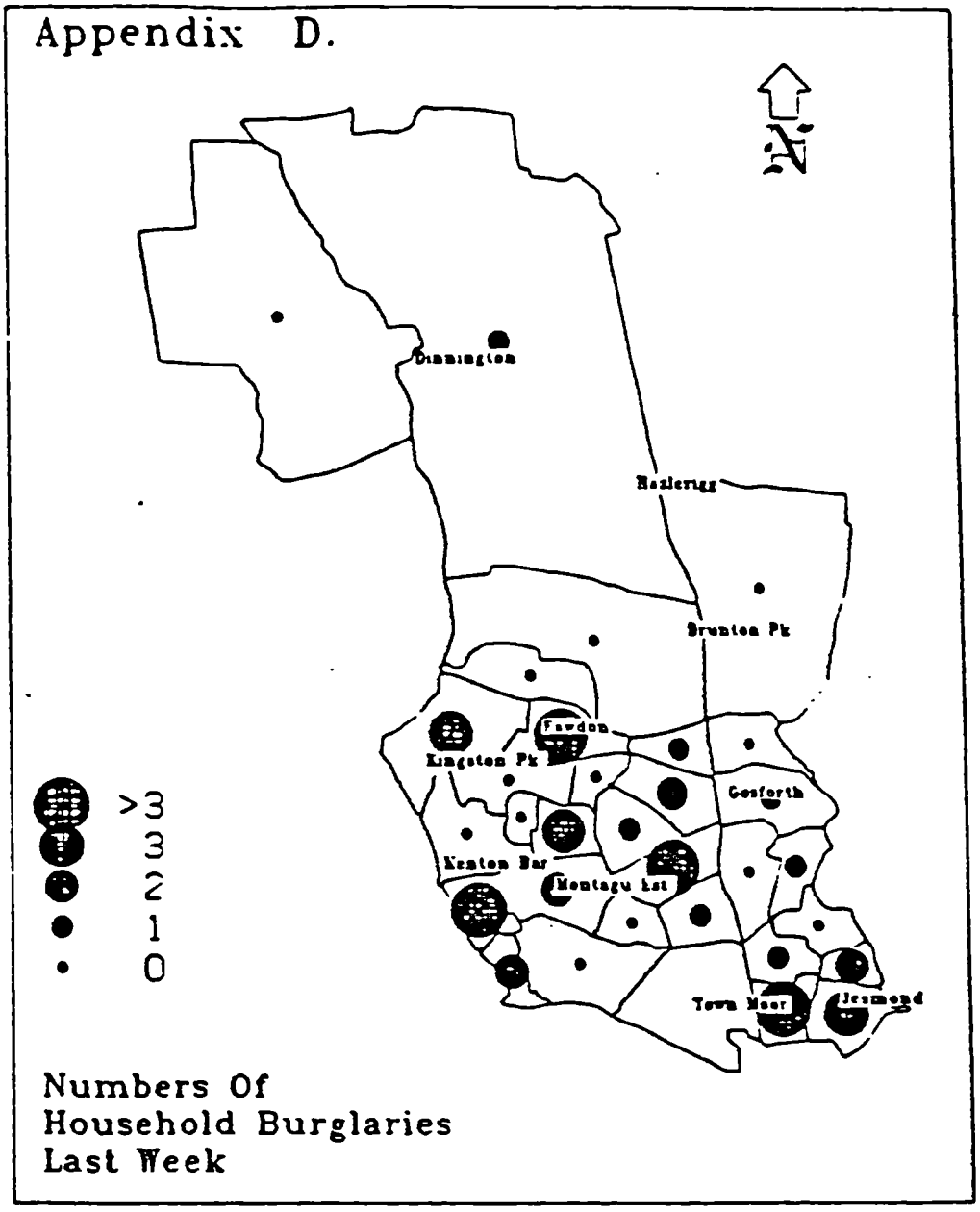

Appendix D.
N
Cunnington
Bruntan Pk
Fawdon
Gosforth
Montagu Est
Town Moor
>3
3
2
1
0
Numbers Of
Household Burglaries
Last Week

Figure 7.4

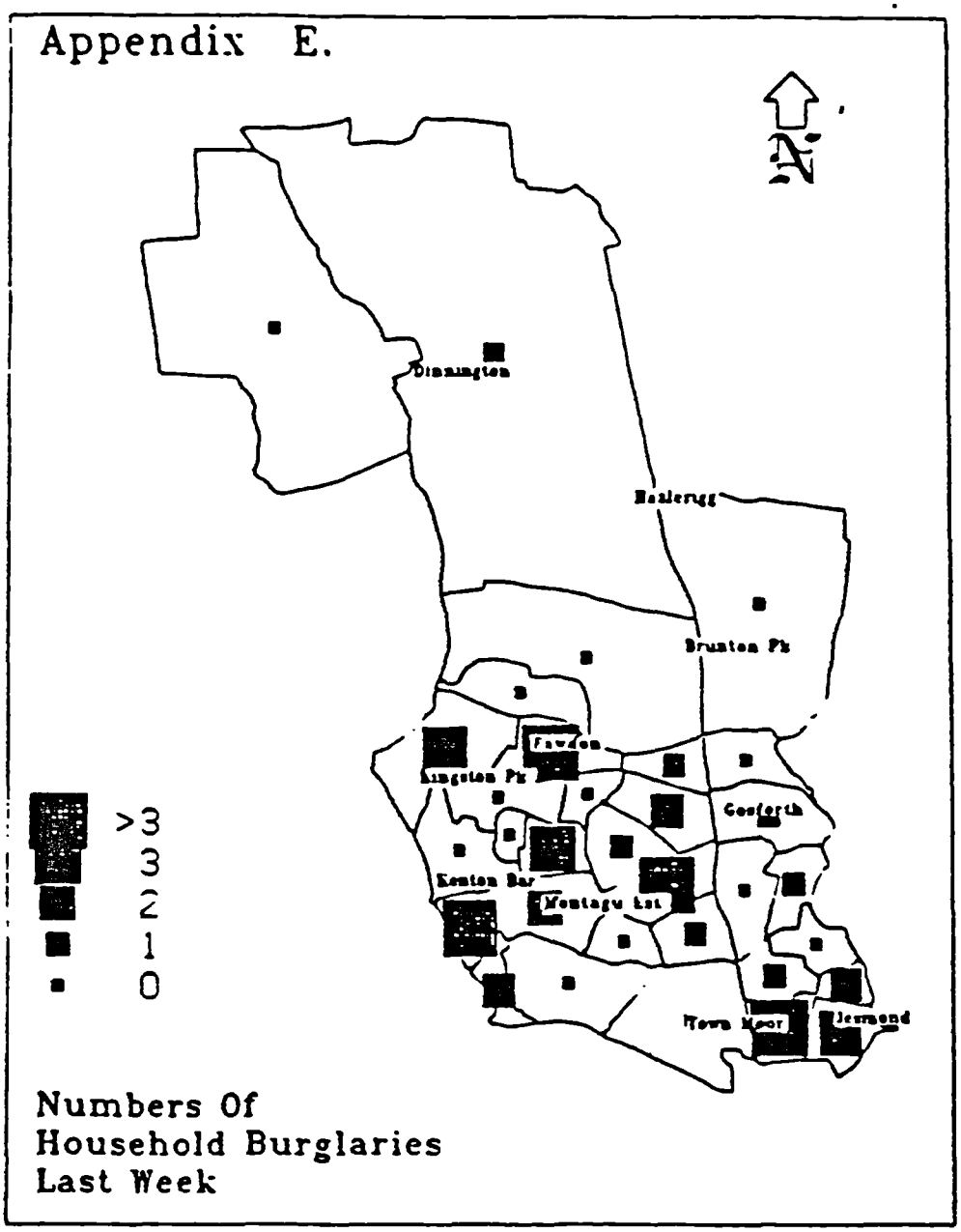

Appendix E.
N
Dinnington
Brunton Pk
Fawdon
Gosforth
Kenton Bar
Montagu Est
Jesmond
>3
3
2
1
0
Numbers Of
Household Burglaries
Last Week

Figure 7.5

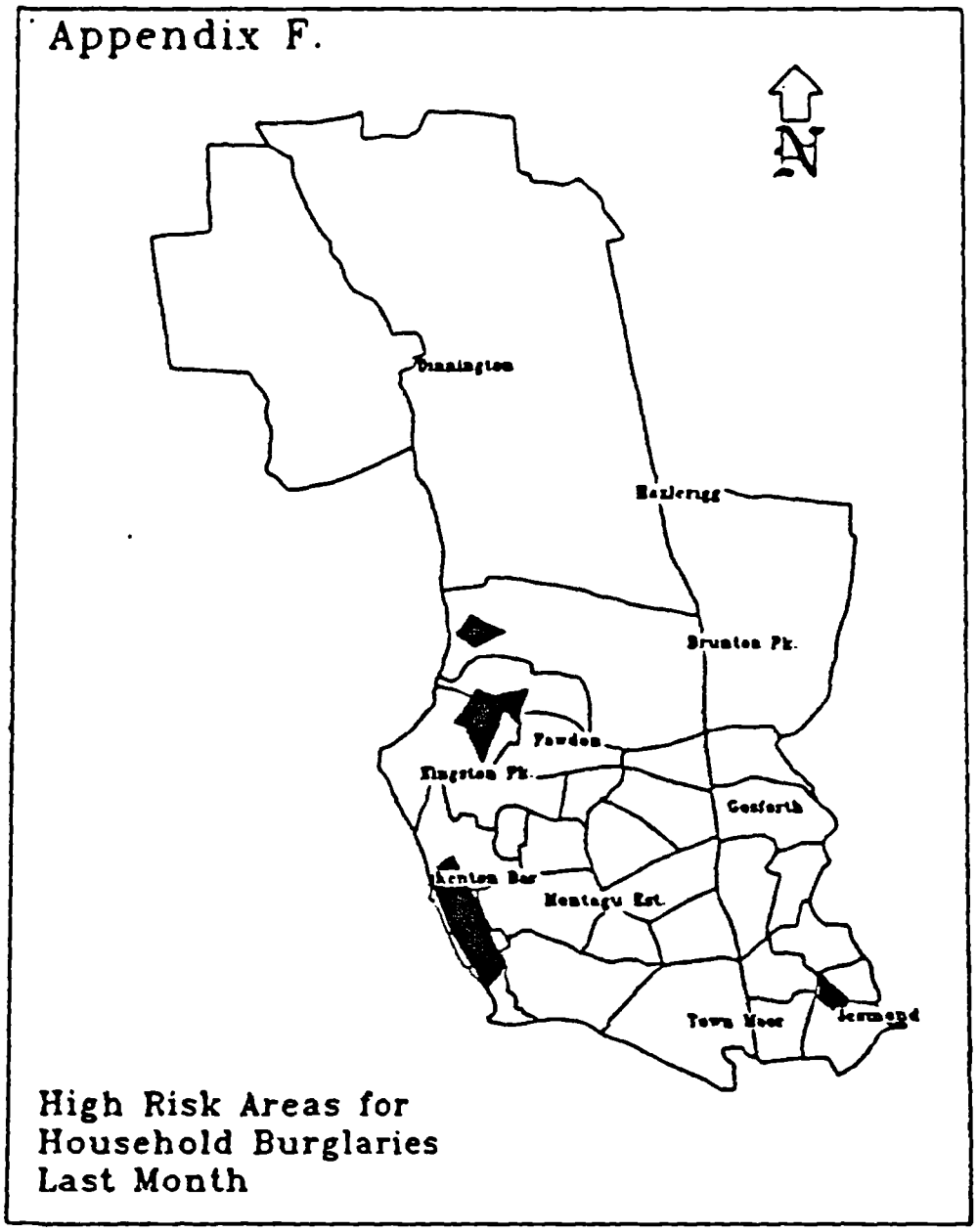

Appendix F.
N
Bexleyrigg
Brunton Pk.
Powdon
Gosforth
Montagu Est.
Town Moor
Jesmond
High Risk Areas for
Household Burglaries
Last Month

Figure 7.6

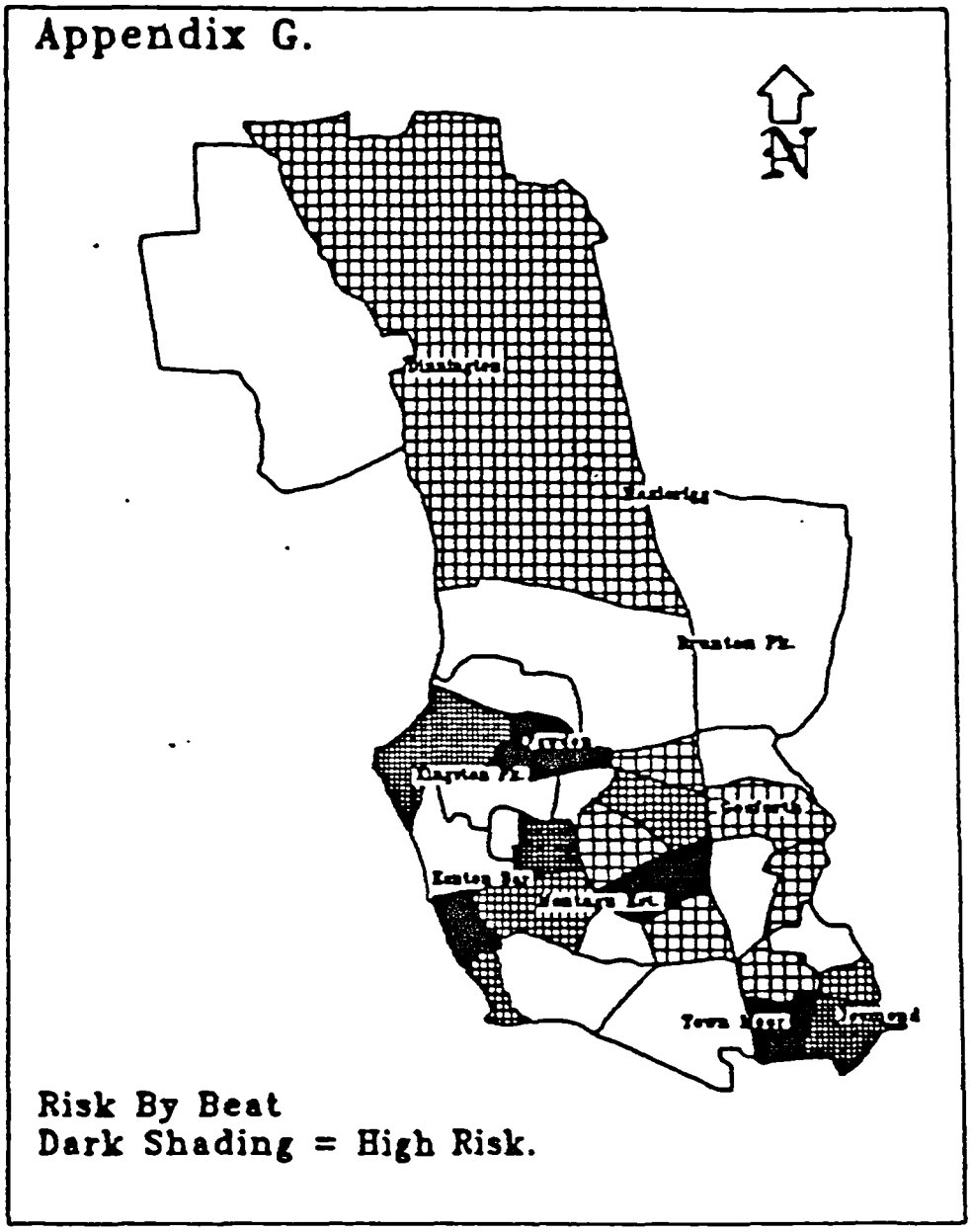


Figure 7.7

7.2.3 Survey Design

The main factor to be examined in the survey is ease of use of the maps. A good map is one that is unlikely to be wrongly interpreted. Thus, the main object of the survey is to determine which of the maps is easiest to understand. The most significant part of the survey should therefore consist of the interviewee's reaction to each of the map formats. For statistical analysis, these reactions should be quantified in some way.

There are two ways of doing this: firstly, the interviewee could be asked to rank each map in terms of 'ease of interpretation', or alternatively he or she could be asked to score each map out of ten for the same criterion. Scoring has the disadvantage that it forces interviewees to calibrate, albeit implicitly, a scale of measurement for the qualitative concept of 'ease of use', and some individuals will calibrate differently to others. Thus, to some people a score of 5/10 may mean 'fair', when to others it may suggest 'poor'. This problem is avoided when ranking is used, but while scoring allows each map to be considered on its own merit, ranking requires evaluation of each map in relation to all others. While this is easy for two or three maps, for larger quantities it is thought that this may be confusing for the interviewee.

It was decided, then, that scoring is the best form of evaluation, as it is a conceptually simpler task than ranking when there are seven different maps. The varying calibration effect could cause problems, but this can be overcome either by analysing scores using two-way ANOVA techniques, which compensate for effects of one factor when

assessing the effect of a second factor, or by replacing the scores by their ranks. In the latter approach, an identical analysis could be performed to that if the questionnaires had asked for ranks, but this approach avoids the need for interviewees to perform the ranking themselves.

In addition to the scoring section of the questionnaire, it was decided to include a comments section. This is partly a device to check for any reaction to the maps which could not be expressed in the scoring section discussed in the last paragraph. It is also useful to pick up suggestions for ways of mapping the data which may have been overlooked in the survey. If a significant number of suggestions of this type are made, this would indicate the need for further research into this subject, probably at a different subdivision.

It also was thought that rank of police officers should be included as a variable in the questionnaire. When the complete system is implemented, it may be that one particular rank of police officer makes use of it considerably more than any other. For example, its operation may generally be allocated to the duty roster of sergeants in each subdivision, and in this case the reaction of sergeants is of specific interest. Also, although other ranks of officer may not have access to the system, they may require output, and if the demand for output is noted to come from one rank in particular, again the reaction of only that rank will be of interest.

The final questionnaire design is shown in figure 7.8. Note that the appendices referred to on this questionnaire correspond to

Questionaire Used To Carry Out Survey Of Police Officer Visualisation Of Crime Incidence Mapping.

Figure 7.8

CRIME MAPPING QUESTIONNAIRE

PART 1

NAME .....

RANK .....

DUTIES .....

LENGTH OF SERVICE .....

PART 2

Please examine the attached Appendices and assign a score to each one between 0 and 10 depending on how effective you consider that type of presentation to be.

For example, if you consider that Appendix A the Contour Map is easy to understand then give it a high score but if you find it confusing give it a low score.

Having decided on the score for each map complete the questions below and return the form to your Sub-Divisional Administration.

Appendix A	Contour Map	score .....?
Appendix B	Incidence Map	score .....?
Appendix C	Crimes per Beat	score .....?
Appendix D	Proportional Circles	score .....?
Appendix E	Proportional Squares	score .....?
Appendix F	High Risk Regions	score .....?
Appendix G	Shaded Beats by Risk	score .....?

PART 3

COMMENTS

Enter here any comments you may wish to make about any of the maps or alternative ways of presenting the information).

figures 7.1-7.7, discussed earlier. A questionnaire was sent to every police officer in the subdivisional headquarters, 112 in all, together with a covering letter explaining the purpose of the survey, and thanking them for their cooperation.
It was hoped that officers' familiarity with the project (the author had worked in the subdivisional headquarters collecting data on several occasions in the past), and the brevity of the questionnaire would result in a good rate of response.

The questionnaires were sent from Northumbria Police Force Headquarters, together with a covering letter, to the administration office for the subdivisional headquarters. From there they were distributed to all officers in the subdivision, who were asked to return the completed forms. After a four week period the returned questionnaires were collected from the same office. After a further two week period, the office was visited once more, to collect any forms handed in later than the initial period. Finally, after a further two weeks the office was visited once more, to collect a final batch of forms. In fact, on this final visit no more forms had been received, and it was decided that those forms collected already were likely to constitute the full response to the survey.

7.2.4 Analysis Of Results

There were 91 responders to this survey, out of a maximum of 112, giving a response rate of 82.7% . Of the non-responders, one had

retired since the list of police officers in the subdivision had been compiled, and one was on sick leave for the duration of the survey response period. The extent to which individual calibration of scoring differs is examined initially. For each individual, the average score over all maps is calculated. This is illustrated in histogram format in figure 7.9. It can be seen that there is a large spread in the mean score given by individuals, which suggests that individual calibration effects must be allowed for when assessing response to the maps. If this were not done, there is a danger that some linkage between 'generosity' and preference for a particular map may result in a misleading conclusion. The compensation may be achieved by performing a two-way analysis of variance on the score data, with effects of individual bias in scoring, and underlying assessment of the maps being estimated. Another approach, as already mentioned, is to replace the scores given by individuals with the ranks of those scores. Both of these approaches will be adopted here.

The results of the two-way ANOVA are shown in table 7.1. Clearly there is statistically significant ($p<0.0001$) evidence for both differing levels of scoring between individuals and between maps. The fitted scoring levels for each map after correcting for individual calibration effects are listed in table 7.2 and shown as a histogram in figure 7.10. Clearly, map format C is the most popular.

A rank-based analysis is also performed. For each individual, the scores given to each map are replaced by their rank (7 for the highest

FIG 7.9 MEAN SCORES AWARDED BY INTERVIEWEES

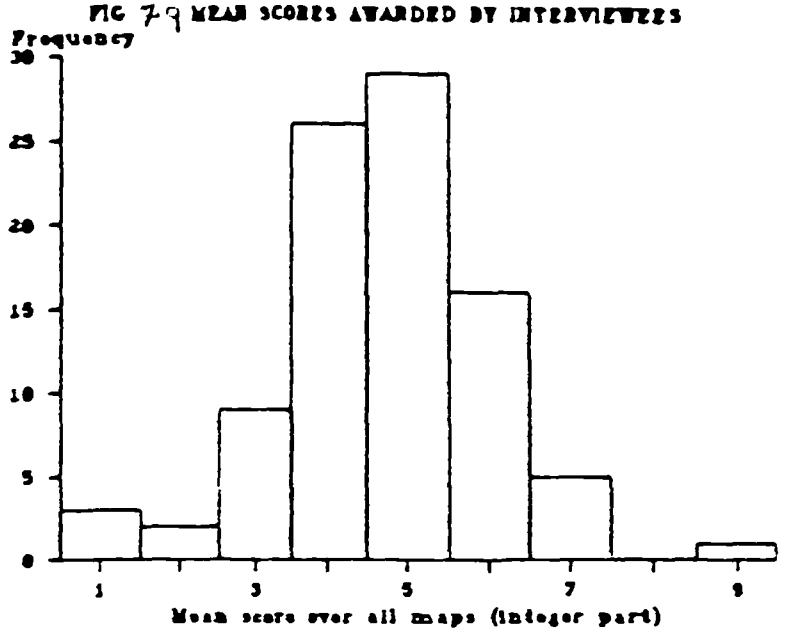


FIG 7.10 MAP SCORES CORRECTED FOR INTERVIEWEE BIAS

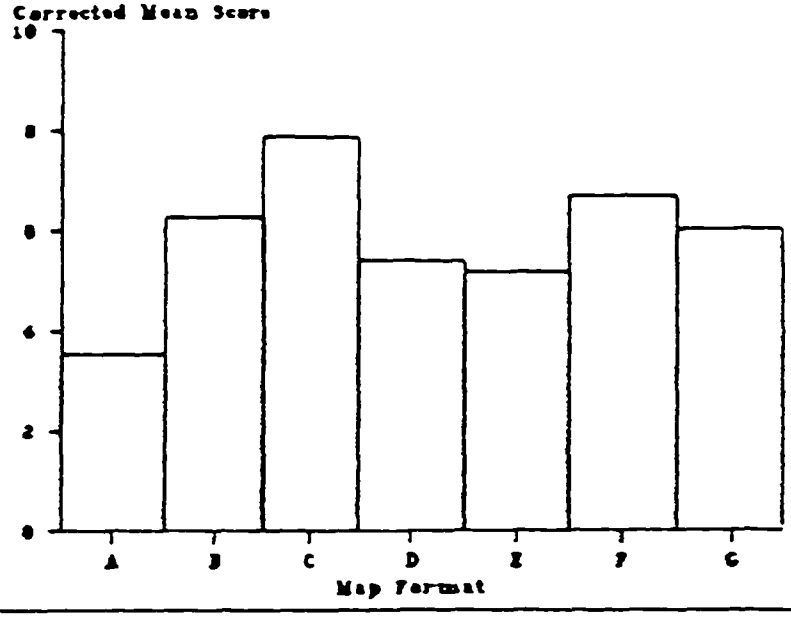


Table 7.1
Two-Way Analysis Of Variance For Map Scores

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F
MODEL	96	2113.16	22.01	4.70	0.0001
ERROR	540	2527.19	4.68		
CORRECTED TOTAL	636	4640.36			

MODEL Term

SOURCE	DF	ANOVA SS	F VALUE	PR > F
INDIVIDUAL BIAS	90	1115.22	2.65	0.0001
MAP SCORE	6	997.94	35.54	0.0001

Table 7.2

Corrected Scores For Maps

Map Format	Score
A	3.540
B	6.265
C	7.870
D	5.408
E	5.188
F	6.694
G	6.035

NB. ANOVA model is GRAND MEAN + MAP EFFECT + INDIVIDUAL BIAS

Corrected score is GRAND MEAN + MAP EFFECT.

score down to 1 for the lowest). Having done this, a Friedman test (1) may be performed. This is a non-parametric equivalent to a two-way ANOVA, used to test whether the rankings of the maps differ significantly between individuals. The results of this test are shown in table 7.3, together with the mean rank for each map. Again, the result is a highly significant ($p<0.0001$) 'between maps' effect.

The above analyses apply to all ranks of police officer. However, it is also required that an analysis of the data split by rank be performed. For each rank, table 7.4 shows mean score ranks for each map, and the significance level for the Friedman test as described above. The number of responders of each rank of officer is also listed. For the lower ranks of Constable and Sergeant, the most popular map format is C. Note that for some of the higher ranks, there were not enough officers to carry out a Friedman test.

7.2.5 Conclusions

When analysing the entire data set, pooling all ranks of police officer, both the two-way ANOVA and the Friedman tests suggested that there were differences in the responses to each of the maps. When assessing the performances of different map formats in terms of ease of use, both the corrected mean scores of maps (table 7.2) and the mean rankings (table 7.4) showed that the most popular map format was C, followed by F and B respectively. This is an interesting result, as these three maps cover the three generic map types discussed in section two. This would suggest that all three

Table 7.3
Friedman Test and Mean Rank For Map Formats

Map Format	Mean Rank
A	2.16
B	4.14
C	5.85
D	3.65
E	3.40
F	4.71
G	4.09

Result Of Friedman Test

CASES	CHI-SQUARE	D.F.	SIGNIFICANCE
91	152.4311	6	.0000

Table 7.4

Friedman Tests Split By Rank Of Police Officers

Rank Of Officer	Mean Rank Of Map Format							No. Of Cases	Signif-icance
	A	B	C	D	E	F	G		
Constable	2.35	4.20	5.70	3.51	3.34	4.75	4.16	61	0.0000
Sergeant	1.75	3.79	6.21	4.43	4.11	4.25	4.16	14	0.0000
Inspector	1.92	4.50	5.83	3.17	3.17	4.75	4.67	6	0.0385

NB. For other ranks of police officer there were insufficient cases to perform a Friedman test.

types of maps are of some use, and that the main distinction made in the scoring was between good and bad formats of each type.

An inspection of the comments given on the questionnaires suggests that clarity of data presentation is an important factor. The less cluttered appearance of map F, the high risk area map, was preferred to the more complicated contour map from which it was derived. Some officers seemed to have difficulty with the concept of crime density contours (or crime risk contours), but most found the idea of high risk areas unambiguous.

Symbolic maps, using varying sizes of squares or circles also proved unpopular. A few users preferred point-based mapping techniques to beat-related representations, some commenting that this highlighted significant pockets of crime crossing borders. It is important to note that all of the three main headings for data representations (beatwise, pointwise and by 'risk surface') all had support in some form, suggesting that a system should represent all of these techniques to improve ease of use for the widest base of users.

When considering the analysis of the data which has been split by rank in the Police Force, it can be seen that for the most common rank in the sample, (Constable) the order of preference for the map formats agrees exactly with that for the pooled ranks analysis. For the next most common rank (Sergeant) there are differences in the order of preference. Although map C is still the highest scoring, map F comes

third, and map B fifth in the rankings. However, it should be noted that the sample size for this category is very much smaller that the other samples considered above, and that variability of results as a consequence of this greatly reduces the reliability of the analysis. For other ranks of police officers, there are so few cases in each of the categories, and Friedman testing will not prove very powerful. Thus, there is insufficient information to draw any useful conclusions for these ranks.

One of the objectives of the final working system is that it will be of aid not only to relatively highly ranked police resource managers but also to lower ranking police officers, who will use this information 'on the beat', and thus the reactions of constables and sergeants are of considerable importance. Both of these ranks have displayed a preference for map type C, and thus it is apparent that a graphical routine producing maps of this format should be incorporated. Although sergeants did not appear to respond well to the point pattern or high risk area maps, constables who are likely to be the principal users of the mapped output from the system showed enthusiasm, and this justifies the inclusion of facilities to draw such maps in the software also.

7.3 An Individual Interaction Study

In this study the working prototype crime system, as specified in Chapter 6, is to be given a "hands on" trial. The purpose of this is to evaluate the design ergonomics and ease of use of the system, as well as

observing the performance of the working system. To best reproduce the conditions of the workplace, a Police Officer who has gained experience in crime pattern analysis has been selected to act as the human user in this trial. Therefore, as the user has a strong concept of the type of tasks that the system has to perform in a practical situation, they will be in a position to make comments relevant to further refinements that could be made to the system, in the transition between prototype and full implementation.

7.3.1 Procedure for Trial

The setup for this trial will now be considered. Since the trial police user will be expected to comment on all aspects of system implementation a situation must be created in which a set of data is to be entered into the system and then analysed. In addition to this, the officer is expected to be relatively familiar with the area for which data is being analysed. For this reason, they were issued with a set of reported household burglaries over a two month period, spatially referenced, and maps of the area. In fact, the officer who agreed to participate in the study had worked in this area some years previously, so that this familiarisation process perhaps took the form more of a "memory refresher" and an assimilation of changes that had occurred in the subdivision since their period of experience of policing the area.

After the familiarisation period, the user was then expected to enter the data into the system. At any time during the entry, they were able to inspect the data using any of the available mapping or tabulation

options. Before actually operating the system, a demonstration was given by the author, showing the user each aspect of the system. In case this was not sufficient to allow fluent operation of the system, the user was then allowed to operate the system using a dummy data set. In this way, they would gain experience of commonly used options for map display and analysis before entering the data.

After this stage, the entry of data was performed. Initially the author supervised the input of data items, in order to explain the operation of the data input system to the novice user. This continued until the user felt confident to continue unaided. After this, the author maintained only a background presence, to handle any major problems or software faults encountered.

Once the data was entered for the two month period, and the user relatively familiar with the crime characterics, the prediction facilities were used, and evaluated against the true crime counts for two weeks following the trail period. This concluded the trial of the system. Throughout the trial the officer was provided with a notepad to write down any comments, either critical or identifying good aspects of the system. At the end of the trial, the user was interviewed, using these comments as a basis for discussion.

7.3.2.1 User Reaction to the System

In this section, the users' comments about ease of use, and suggests as to improvements which may be made in the system are considered. The

interview with the user raised certain points about the design of the system, and the main conclusions reached between the user and the author are outlined in the following sections.

7.7.3.2 Crimes External to the Subdivision

When crimes are entered into the system database, they are spatially referenced by postcode. The postcode is then converted into a grid reference by a look-up table. Clearly, as the crimes are only recorded over the extent of the subdivision, look-up values are only provided for postcodes falling within the subdivision. If a crime has a postcode that is not in the list then it cannot be analysed, and so is not recorded in the database.

During the system trial, however, it was found that some members of the public reported household burglaries occurring just outside the subdivision, or just on its borderline. This is hardly surprising since the borderlines delimiting the areas served by subdivisions are not generally known. As a result of this, however, some crimes could not be entered into the database. Although these crimes would not feature in a tabulation of crimes by beats, they may be of use when detecting Knox clusters (see Chapter 4) or when examining point patterns.

It was felt by the user that, rather than excluding this type of observation, they should be included in the database. This an important point: in chapter 4 (section 2) it was discovered that clusters of crimes often straddled beat boundaries, and that only examining processes on

one side of these boundaries may obscure detection of some patterns. This must also be the case with subdivisional boundaries, which are equally artificial.

A suitable response to this would be to buffer the area of the subdivision, by about 1km, and to store lookup values for all postcodes within the expanded regions. In this way, occurrences just outside of the subdivision may be stored in the database, plotted onto visual map displays, and used for analytical purposes.

7.3.2.3 Batch Input of Data

The current software for the entry of data into the system presents the user with a full-screen form, with boxes into which data may be typed. Although this was thought easy to use, it was felt that after a while, when the user was familiar with this type of input, that this method of data entry was too slow. It was thought that there would be times when several data items would need to be entered in a block. Currently, each item requires a full screen form-type entry. During input, error checking, postcode verification and data collation takes place. A result of this is that there is a delay between data items being entered, and during this period the officer entering the data is "held captive". It was thought that, particularly when a large number of incidents had to be entered at a single sitting, this would be time-consuming and discourage the user. It was felt also that if the user were discouraged from data entry, their enthusiasm for using the analytical and mapping aspects of the system may also be diminished.

A proposed solution to this problem may be to allow the user to compile a "block" of data entries, using a screen editor, without verification, which may then be fed into the database as a batch job. As each record is read, it may then be checked, and a list of records failing to qualify for database entry could be provided, allowing user modification later on. In this method of entry, once the text file has been compiled, the batch entry of data could be executed without supervision. Thus the manpower overhead would be reduced.

It is hoped that eventually data will be read from a communication link with a forcewide central database (see chapters 6 & 8) in which case, local data input will not be required at all. However, in the interim period the above method of batch data entry may prove more practical than the initially proposed form-filling procedure.

7.3.2.4 Rolling Prediction Horizons

The system currently forms its predictions on a Saturday-to-Saturday basis, allowing a prediction to be obtained once a week, at the beginning of a week. However, it was thought that police resource managers may require forecasts at other times during the week, in addition to this.

It was thought helpful, therefore, if the system were able to make predictions for the period of seven days beyond the current date. This may provide difficult to implement, although it could either be done by pro-rating the remaining part of the current week's prediction, and the complimentary part of the subsequent week, or by re-forming the

week-by-beat cross-tabulations on each working day, so that the weekly categories are based on seven day intervals terminating exactly at the current date; from a table of this format a rolling, seven days ahead prediction may be made.

7.3.2.5 Postcode Entry Correction

The problem here relates again to the post-code verification routine. When an erroneous postcode is entered, and fails to be found in the look-up table, then an error message is displayed and the user given an opportunity to re-enter the postcode. This method is effective when a correct postcode is mis-typed, but if the postcode is correct, but refers to an event just outside of the study area, then no correction can be applied. In this case the user would become stuck in an infinite loop, unless they use the escape sequence. This was felt not to be sufficiently user friendly. A remedy for this may be to set up an option for the operator to abandon a record after, say, two unsuccessful attempts to input a postcode.

7.3.2.6 The Comment Option on the Point Map

This was the only criticism aimed at data presentation rather than input. When displaying past data as a point-mapped option, one of the facilities available is to display comments associated with individual crimes, which have been entered as a part of the household burglary input procedure. This was found to be useful for linking up subjective data about the burglaries with their spatial patterns.

The current means of accessing this information is by firstly positioning a small cursor over the crime in question, and then pressing a further button to obtain the comment text. The cursor is not moved in the usual up, down, left and right control key format, but jumps between the crime points in reference number order. The means of moving the cursor was found to be useful (mainly because crime pattern analysts are used to thinking in terms of the crime reference number), but it was felt that, rather than having to press a key after identifying the crime to examine, it would be better if the text wee displayed automatically. Thus, as the cursor moved from point to point on the map, the text window would simultaneously change its comment. In this way, users could either be seen through crimes in a spatial sense, by watching the map, or scan through the comments, looking for key words or phrases, and then identify the events spatially.

7.3.2.7 The Menu Based System

Having discussed the criticisms of the system precipitated in the trial, some consideration will now be given to points that were found praiseworthy in the system. The principal of these is the menu system. The officer involved in the trial commented that they felt "confident" and "in control" of the software, after having tried to deliberately press key options not offered on a menu screen. Having tried this, they felt that it was unlikely to damage data, or the system hardware, by wrong key presses, and therefore felt encouraged to experiment with the system and explore its facilities.

7.3.2.8 The Choice of Analysis and Display Formats

A final, and general point made by the officer involved in the trial was that they felt that the large choice of map display format, and analysis method was an important positive factor. They felt that personally, they found the forecasting, and Knox cluster options most useful, but that other crime pattern analysts may prefer, for example the identification of high risk areas, using kernel estimators (see chapters 4, 6).

The important point is that several options are on offer, allowing users to develop personal methodologies for examining crime. Some analysts perceive pattern information in different ways to others, and if all are to gain useful output from the system, and take appropriate corrective action, then a wide variety of techniques should be available on the menu system.

There is a parallel between the above, viewed as a subjective observation, and the map visualisation study earlier in the chapter, where several formats of geographical data map types all had significant support from the police officers in the questionnaire survey.

7.4 System Performance

Before considering the prediction stage of the trial, in terms of user interaction with the machine, the purely data-based prediction results will be considered. A typical week of predicted values of weekly crime counts for the period during May 1984 is listed in table 7.5. The

Table 7.5
Predictions Of First Week After Bi-Monthly Learning Period

Beat Code	Prediction		Outcome
	Police	Machine	
T1	0	1.2	2
T2	0	0.4	0
T3	0	0.4	1
T4	0	0.4	1
T5	1	1.0	0
T6	1	0.7	0
U1	0	0.4	0
U2	0	0.4	0
U3	2	1.3	1
U4	1	1.1	0
U5	1	0.8	0
U6	0	0.4	0
V1	0	0.4	1
V2	0	0.4	1
V3	1	0.5	0
V4	0	0.4	0
W1	0	0.4	1
W2	0	0.4	0
W3	1	0.5	1
W4	2	0.9	0
X1	1	0.4	0
X2	2	0.6	1
X3	2	0.6	0
Y1	0	0.4	0
Y2	2	1.0	1
Y3	2	1.7	1
Y4	1	2.2	2
Y5	1	0.8	1
Z1	2	1.4	4
Z2	3	2.3	7
Z3	4	2.5	4
Z4	0	0.4	0

predictions are based entirely on the stochastic model of chapter 4, and exclude the subjective, Bayesian type of input as described in chapter 5. It is noted that generally the predictions perform reasonably well (usually out by only 1 crime) except when there are particularly large crime counts in individual beats (say 5 or more).

It might be thought that the statistical model manages to explain the "background" process, but that occasionally a surprisingly high amount of burglaries occur in a particular area: this cannot be foreseen in past data. This was also felt to be a reasonable model by the police crime pattern analyst user. Often, there will be substantive explanations for the sudden "crime waves" occurring, but these may not be detectable in the past data.

It is difficult to evaluate the effectiveness of the Bayesian element in the system, outside of the full model implementation, since it is difficult to simulate the local knowledge of the crime pattern analyst at the time that the archived events were occurring. However, using maps, and also examining modus operandi details from this past data, the analyst attempted to use the Bayesian prediction facility of the model as though the analysis was occurring in real time, and the analyst had the corresponding subjective knowledge.

Using this technique, a set of modified predictions were obtained, which are shown in table 7.6. These illustrate certain situations where the human analyst was able to spot certain patterns which the past data

Table 7.6
Combined Predictions

Beat Code	Combined Prediction
T1	0.9
T2	0.2
T3	0.2
T4	0.2
T5	1.0
T6	0.9
U1	0.2
U2	0.2
U3	1.4
U4	1.0
U5	0.9
U6	0.2
V1	0.2
V2	0.2
V3	0.9
V4	0.2
W1	0.2
W2	0.2
W3	0.9
W4	1.7
X1	0.5
X2	1.0
X3	1.0
Y1	0.2
Y2	1.5
Y3	1.9
Y4	1.7
Y5	0.9
Z1	1.9
Z2	4.0
Z3	3.6
Z4	0.2

alone was unable to detect (such as, in one beat when a number of similar crimes occurred in the latter part of the week).

7.5 Conclusions

The intended users of this system have been considered in this chapter. Firstly, a large scale survey (over a subdivision) gave some insight into the types of data representation that are most effective at communicating spatial information relating to crime patterns. It was found that there was notable support for each of the three main map types identified in section 2. In the crime prediction and analysis system all of these formats are offered; it is hoped that this flexibility will allow diverse analysts in various subdivisions, to tailor the system to their own needs.

The results of the single user trial identified that most of the aspects of operation that required alteration were in terms of data input. There was only one criticism relating to map format; in fact the crime pattern analyst remarked that they found interactive map analysis options both easy to understand, and simple to use. Ultimately, data input will not be problematic as the system will be fed from an external database, but in the mean time, the suggested improvements may be relatively easily effected.

On a final and more general note, it was stated in the introduction to this thesis that computerised crime pattern analysis was to be investigated, with a view to a practical implementation. This cannot be claimed unless, in addition to the mathematical modelling and computer

implementation aspects, the end result is a system which may be easily used by those police officers requiring the information it has to offer. It has now been demonstrated that a police officer experienced in crime pattern analysis is capable of operating the system without major difficulty, and therefore, it is reasonable to expect the concept of a subdivisional based computer system aiding in the analysis of crime patterns to become a reality.

C H A P T E R 8

CONCLUSIONS AND POINTS OF DEPARTURE

8.1 The Introduction of Automated Analysis to Crime Data

It was stated initially that a principal aim of this thesis was to investigate methods of spatial pattern analysis that may be applied to local crime data. In chapters 4 and 5, statistical techniques were considered which may be applied to the occurrence of crime as a geographical process. In the exploratory analysis of chapter 3, other non-geographical techniques were also introduced. All of these could be applied to local crime data which may be routinely recorded at a police subdivisional level, and so provide a means of analysis that may be easily realised. In Chapter 6, methods of implementing such techniques on a micro were proposed.

It is important to note that while the ultimate goal of chapter 5 was a crime prediction system, the preliminary spatial statistical tests that were used to analyse patterns in the data yielded useful techniques in themselves. For example, the knox tests in chapter 4 can be made to highlight local "clusters" of household burglaries, which may then be mapped. Considering the viewpoint of the crime pattern analyst, as described in the introductory chapter, an important group of household burglaries may be identified automatically. In the manual case, or even the case where the analyst has access to database software, the

detection of pattern has to be based on inspection of raw data. In this case, the likelihood of error is high, and the task is time-consuming.

The above example is based on identifying the individual crimes thought to be important. In addition to this, Kernel estimation techniques enable regions at high risk from crime over a long period of time to be identified. As the technique is based on point referenced data rather than areal units of aggregation, areas of high risk straddling beats may also be identified. Again, for the analyst this is important: foot beat officers assigned to a particular beat often may not observe events in adjacent beats, and part of the task of the analyst is to identify patterns of this kind.

In addition to the application of the spatial techniques, the analysis of the time-of-day data is also of use: deployment of police resources at different times of day (or seasons of the year) may depend on areas in the locality being subject to differing risks. For example, burglary would appear to occur only extremely rarely on households between 5.00am and 9.00am. In this case, foot patrol officers may be briefed to concentrate on other crimes more likely to occur during these hours.

Thus, spatial analytical and other techniques could contribute to the set of tools available to the crime pattern analyst, performing the repetitive, pattern scanning tasks and allowing them to concentrate on intelligent, but more subjective analysis of the emergent patterns. Finally, using the Bayesian techniques set out in chapter 5, the results of these analyses could be re-combined with the statistical pattern projection.

8.2 Realising a Full Working System

A prototype crime prediction and pattern analysis system has been created from the work of this study and such a system has been found to be useful by crime pattern analysts. It must now be considered how such a system could eventually be implemented as a full working system.

8.2.1 Hardware

From a hardware viewpoint, implementation is not particularly problematic. The pilot system has been developed to run on an IBM PC compatible computer, and this has become a widespread de facto standard for personal computers (see chapter 6). It is possible, however, that additional hardware may eventually be required after purchasing a basic model. This will be to accommodate the increased graphical detail required if the system is linked to a geographical information system (see 8.2.2 & 8.2.3), and also to accommodate possible increases in data storage requirements, either for the above reasons or for other software enhancements to be proposed later in the chapter. It may also be preferred if the final working system can drive a plotting device of some type, for the production of hard copy of maps of crime patterns, predictions and so on.

8.2.2 An External Data Source

At the time of designing the prototype, although centralised recording of incident reporting had been established in the Northumbria Police Force,

the central database facility did not store the postcodes of incidents. However, improvements in this are currently being developed, and it is expected that a postcode recording system will be implemented by 1992. In addition to this, the system will consist of a central file server, which will download requested data to micros at subdivisional level. It is also proposed that this data will be readable by other software packages.

The prototype system developed on this study required its own data entry system, mainly as the postcoding of data (necessary for spatially referencing) was otherwise not recorded. In the future, however, as this data will now become centrally available, it seems reasonable to read data into the analysis system from the file server. This obviates the need for data to be entered twice, and allows more sophisticated filtering of the data for analysis, which may be performed by the central database software currently being developed in SQL (Structured Query Language - a database definition language).

Since the system has been designed on a modular, menu-based model, adaption to this should not be difficult. The current data entry program could be replaced with a program capable of communicating with the central file server, and reading in appropriate data for analysis using the techniques already implemented. Once this has been done, alteration of the menu descriptor files to allow for this could replace the new data reading module in the place of the original user data entry module in the overall system.

8.2.3 Linkage to Geographical Information Systems

Another direction in which the system may be expanded is in its graphical and cartographic output. The prototype system gives fixed-scale mapping showing beat boundaries as background data, onto which choropleth shading, risk contours or point data may be overlayed. However, output of this type may be fed as data into a Geographical Information System package, after analysis has been performed. This allows the results to be examined in a more informative geographical context. Added to the background information could be OS maps, and positional data relating to various police specific attributes neighbourhood watch areas, for example. With most GIS packages, the facility to "zoom in" on particular sections of the map is also available.

Thus, in the case of, say, the Knox cluster detection, a detailed view of a street were several burglaries have occurred could be obtained. This may show relative positioning of houses, location of back allies and other access points, and further features, allowing the analyst to look for further connections and similarities in the crime patterns. As in the last section, certain modules in the system could be altered to output results into a GIS rather than directly onto the screen.

A further advantage of this type of approach is that, since the crime pattern analysis system becomes less hardware specific, possibly communicating with a GIS in a standard format (possibly ASCII), if the main hardware were to be altered at a future date, to implement a faster

or larger system, then if the software is written in a standard programming language there should be little difficulty in loading it onto the new hardware.

8.3 Further Development of Analytical Techniques

In addition to the software extension discussed above, some of the central analytical techniques may also be taken in further directions. Although the system as it stands provides the crime pattern analyst with a set of tools, there are ways in which some techniques may be adapted to be used in different situations, and, with the advent of GIS systems, there are techniques which may allow crime pattern analysis as described here to be combined with map data to provide further geographically oriented descriptions of crime data.

8.3.1 Space Time Prediction Models on A Force-Wide Scale

The space-time autoregression methodology used for forecasting weekly crime rates on a foot beat scale of aggregation may also be applied to geographical data on a larger scale. In this context, the analysis could be used at force headquarters level as a management means of allocating resources between subdivisions over, say, yearly periods. In this context, predictions could not be made in a Bayesian framework (since the type of local knowledge that beat police officers could apply to small scale crime patterns would not be available here). However, space-time autoregression models may be calibrated in a more orthodox sense, and may be used as a basis for prediction.

8.3.2 Extention of the Concept of "Risk Surfaces" to a GIS Environment

The technique of Kernel estimation described in chapter 4 (Silverman, 1983) provides an estimate of a function of two-dimensional space, mapping grid co-ordinates onto a crime "risk surface". As illustrated in that chapter, and also in the implementation of the prototype system, these provide a useful mapping facility. Incorporating the ideas of multivariate calculus (see for example Kolman and Trench, 1971) into this framework, many other useful geographical descriptive methods may be derived.

If the surface is thought of as risk density, then, for an arbitrary area within the subdivision, the crime risk inside the area can be thought of as the volume beneath the risk surface if this area is extended upwards (see figure 8.1). This may be represented by the volume integral

$$\iint_A r(x,y)\,dx\,dy$$

where $r(x,y)$ is the risk density function, and A is the area over which x and y vary. For an arbitrary area, this integral could not be evaluated analytically. However, numerical approximation techniques could be applied if the value of was given over a grid of and

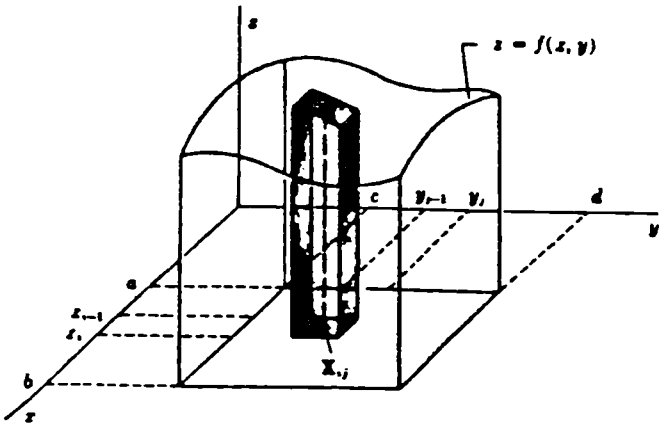


Figure 8.1

values. This could be used in a GIS context. The total crime risk could be evaluated for an arbitrary polygon drawn over an OS map on a VDU using a mouse. This allows the extension of the idea of identifying high risk areas to that of giving the risk associated with an arbitrary region. This could be expressed either as an absolute quantity, or standardised to unit area. This may prove to be helpful if a particular estate, or part of a locality was believed to be a "problem area". As with the other techniques in the system, the concept of area integrals would be hidden in the software, and the front end would present the problem in terms of local crime geography.

As well as area integrals, line integrals could also be applied. A line integral would evaluate the total risk along a single dimensional path within the subdivision, if the path were parametred as $(x(t), y(t))$

then the integral would be

$$\int_{t_0}^{t_1} r(x(t), y(t))\,dt$$

as before this could be approximated numerically, given the path as a list of vertices, and either known values of at these vertices, or a grid of values allowing interpolation. Integrals of this type could be used, for example, to measure the risk of particular streets within the subdivision. This would allow a GIS to identify and map the "most susceptible streets", possibly producing a "league table".

Another interesting application of this might be in terms of beat boundaries. In chapter 4, the problem of high risk areas overlapping beat borders was identified. This technique may be used to select and map those beat boundaries with the highest integrated risk factors. It seems reasonable that those boundaries having high risk may cut through two-dimensional regions of high risk.

The lack of speed at which micros are able to perform computations of the type proposed here may currently be prohibitive, but it is important to note that recently available floating hardware will be of aid here. As an example, Intels recently announced i860 cpu may be used as a second processor, and is capable of performing 3D graphics computations as of its basic instruction set. (see Personal Computer World, July 1989). These are primarily intended for CAD applications, but surface interpolation related instructions could be adapted to computations required here.

This also delivers further opportunities to improve risk surface estimation techniques. For example, fault lines could be included, so that risk surfaces could have discontinuity, possibly for household burglary on differing sides of a railway line, or motorway. Initially computation would be performed with user-specified fault lines, although at a later stage edge-detection techniques could be employed to find these automatically.

8.3.3 Improvement of Space-Time Models

The advent of improved floating point hardware may have implications in the prediction software also. In chapter 5, it was stated that certain correlations had to be excluded from prediction models, since this would require the inversion of large matrices. This predominantly affects the predictive distributions supplied for future weeks. However, if hardware speed is significantly improved, it seems reasonable to investigate numerical matrix inversion techniques which may be used in conjunction with this hardware, to give an overall improvement in the space-time modelling used as a basis for prediction.

Alternatively, improvements in the model could be made by increasing the geographical detail. For example, instead of forecasting on a foot beat areal unit system, it may be possible to work in terms of postcode sectors at some point in the future. As suggested earlier, this is not currently feasible, but as above, the advent of faster hardware may provide the potential to do this.

8.3.4 Bayesian Prior Construction

It is important to incorporate subjective prior knowledge into a crime prediction system. However, as discussed in the introductory chapter, and also in the implementation chapter, although the Bayesian framework requires the input of subjective prior beliefs to be represented as a

probability distribution, it is unreasonable to expect crime pattern analysts, whose expertise does not lie in the field of statistics, to supply information in this form. Thus, methods were considered for "hiding" the mathematical aspects, by asking questions about the risk of each beat, and then asking users to supply a degree of certainty that could be applied to their subjective forecast. However in future developments, firstly the number of areal units may increase, and also the complexity of the forecasting model may increase. If this occurs, whereas previously specifying beatwise risk priors may have easily been carried out by directly selecting individually, this may not be practical with small units, such as postcode sectors. In this case, more sophisticated means of constructing prior belief distributors over a set of spatial units may be required.

This could be done using an expert system to ask questions and, possibly by linking local area and street names with point references, build spatial distributions. Some of the ideas behind diagnostic systems such as MYCIN (Barr and Feigenbaum, 1982) could be used to identify areas that were though to be of high risk. Questions could be asked in terms of odds, or betting.

An alternative approach may be to use a mouse interface, and ask the user to identify areas thought to be at risk, and then to return to text questioning to find out degrees of confidence, again perhaps in terms of betting odds. Obviously, the amount of risk aversion will vary between users, but if calibration of the priors supplied takes place, this may be compensated.

8.3.5 Incorporation of Bayesian Methodology Into Risk Surface Estimation

It has already been illustrated that the risk surface techniques may be useful in a GIS context. However, in many cases, the empirical determination of these surfaces will only be as reliable as the data that they are based on. In this case, non-reportal of crimes may distort the data base, so that estimates of risk surfaces may be unrepresentitively low in some areas. it is also possible, given the hypothesised causes of non-reportal of crime (Walker, 1983), that under representation may be concentrated in certain areas. Attitudes towards the police and crime reporting may vary between neighbourhoods, and in certain communities non-reporting of crime may be considerably more common than in others.

Despite the fact that crime data itself may not provide a full description, it is possible that in conjunction with local subjective knowledge, a fuller picture may be obtained. As with the prediction problem, a Bayesian technique may provide help. A technique could be developed in which prior knowledge of the chances of an occurrent crime being reported, as a function of area, may be combined with point referenced incidence data to yield a risk surface (Lo, 1984). The prior distribution could either be derived on a subjective basis, using methods set out in 8.3.3, or an empirical estimate of non-reportal probabilities could be derived from a questionnaire of local communities. Other factors effecting this, such as investigating which areas have neighbourhood watch schemes, and how

successful these schemes have been, could also be incorporated into the prior distribution.

8.3.6 Controlling For Response

An interesting problem arises if the system is implemented in some subdivision and successfully identies problem areas. If police resources are targetted successfully at these regions, and potential crimes are prevented, this may then cause the predictions to become wrong! Clearly, these may have repercussions at some stage in a self-regulating system. Two possible methods of combating this are possible. Firstly, when monitoring predictions for success, weighting of penalties for under-prediction could exceed those for overprediction. Thus the kind of error described above could be allowed for, and not seen as requiring as much attention as an underprediction. Alternatively, if there were some means of manpower resource monitoring over space, this could act as a geographical control variable, so that the measure of crime risk would then become successful crimes per man-hour of policing in a region. If this were the case, then predictions could be made in terms of areas requiring more resources, rather than areas likely to have high crime risks.

However, an option of this sort would have to be offered alongside that of the risk identification and cluster analysis options, as the system should be informative not only to police resource managers, but also to beat policemen. The latter would still find the identification of hogh

risk areas and 'suspicious' crime clusters in there own beats to be important information.

8.4 Concluding Remarks

In this study, the purpose was not to derive criminlogical or behavioural explanations of crime processes, but to derive quantitative methods which may be used to examine empirical crime patterns on data which is readily available to the police force at subdivisional level. This work is necessary for a crime pattern analyst since before considering the causes of the patterns, these patterns must be identified. Taking this one stage further, the aim was then to use these empirically examined regularities as a basis for short-term forecasting. It was found that this could be done, but that the method could be further improved if knowledge beyond the scope of the database could also be assessed.

Although the prediction model may be viewed a an ultimate goal in the project, the analysis of the data both in the initial, non-geographical exploratory context, and then in exploratory analysis of the data as a realisation of a random process in space and time yielded other useful techniques. These may also be incorporated into crime pattern analysis software; for example, time of day profiles for crime are useful information for subdivisional resource managers.

Other "spin-off" techniques such as the Knox-test type of analysis and the Kernel estimations of crime risk were also found to be useful tools. These techniques analyse spatially referenced data and their outputs are

essentially geographical data entities. Thus, they may be input into GIS software, and users may view the results in the context of local geography. Given the applicability of these techniques (perhaps not only in terms of crime), and the recent advances in GIS technology, the further analysis of these may be an important point of departure.

The techniques used have also proved to be relatively easily expressed in non-mathematical terms, although their operation is essentially mathematical. Thus, the police officer may treat them as a "black box" asking and answering questions in terms of crime patterns, rather than mathematical or probabilistic theory. Results may therefore be more easily interpreted in terms of police manpower management and phenomena relevant to local policing. Linked with other, non-geographical data, they provide the crime pattern analyst with information which they may use to identify and interpret incoming crime data more reliably, and faster, than would otherwise be possible.

BIBLIOGRAPHY AND REFERENCES

Aldred B.K.
"Points In Polygon Algorithms"
UKSC-0025, IBM Scientific Centre, 1971

Ambramowitz M. and Stegun I. A.
"Handbook Of Mathematical Functions"
Dover, New York, 1972

Anderson O.D.
"Time Series Analysis and Forecasting: The Box-Jenkins Approach"
Butterworth, London, 1976

Arnold S.F.
"The Theory Of Linear Models and Multivariate Analysis"
Wiley, New York, 1981

Atkinson K.E.
"An Introduction To Numerical Analysis"
Wiley, New York, 1978

Barnett V.
"Comparative Statistical Inference"
Wiley, New York, 1982

Barr A. and Feigenbaum E.A.
"The Handbook Of Artificial Intelligence, Volume 2"
Kaufman, New York, 1982

Bartels C.P.A.
"Operational Statistical Methods For Analysing Spatial Data"
Exploratory and Explanatory Analysis Of Spatial Data
Bartels C.P.A. and Ketellaper (eds), Martinus Nijhoff, Boston, Mass.

Bartlett M.S.
"Statistical Estimation Of Density Functions"
Sankhya (A) 1963, Vol. 25, pp245-254

Barton D.E. and David F.N.
"The Random Intersection Of Two Graphs"
Research Papers In Statistics, David F.N. (Ed), Wiley, New York, 1966

Baxter R.S.
"Computer and Statistical Techniques For Planners"
Methuen, London, 1976

Bennet R.J.
"Spatial Time Series"
Pion, London, 1979

Berger J.O.
"The Robust Bayesian Viewpoint"
The Robustness of Bayesian Analyses,
Kadane J.B. (ed),
North-Holland, New York, 1984

Besag J.
"Efficiency Of Pseudo-Likelihood Estimators For Simple Gaussian Fields"
Biometrika, 1977, Vol 64. pp616-618

Besag J.
"Spatial Interaction and The Analysis Of Lattice Systems"
Journal Of The Royal Statistical Society (B) 1974, Vol. 36, pp192-236

Besag J.
"Statistical Analysis Of Non-Lattice Data"
The Statistician, 1975, Vol 24, pp179-195

Besag J. and Diggle P.
"Simple Monte Carlo Tests For Spatial Pattern"
Applied Statistics 1977, Vol 26. pp327-333

Birkoff G. and Maclean S.
"Algebra"
MacMillan, London, 1967

Bishop Y.M.M., Fienburg S.E. and Holland P.W.
"Discrete Multivariate Analysis, Theory And Practice"
M.I.T. Press, Cambridge, Mass., 1975

Box E.P.G. and Tiao G.
"Bayesian Inference In Statistical Analysis"
Addison-Wesley , Philippines, 1973

Box G.P. and Jenkins G.M.
"Time Series Analysis: Forecasting and Control"
Holden-Day, New York, 1976

Bresenham J.E.
I.B.M. Systems Journal 1965, Vol. 4, pp25-30

Burrough P.A.
"Principles Of Geographic Information Systems for Land Resources Assessment"
Clarendon, Oxford, 1986

Carr-Hill R.A. and Stern N.H.
"Crime, The Police and Criminal Statistics"
London, Academic Press, 1979

Cliff A.D. and Ord J.K.
"Spatial Autocorrelation"
Pion, London, 1973

Cliff A.D. and Ord J.K.
"Spatial Processes: Models and Applications"
Pion, London, 1981

Coleman C.A. and Bottomley A.K.
"Police Conceptions Of 'Crime' and 'No Crime'"
Criminal Law Review, 1976, pp 344.

Conte S.D. and De Boor C.
"Elementary Numerical Analysis - An Algorithmic Approach"
Mc.Graw Hill, New York, 1980

Cox D.R. and Oakes D.
"Analysis Of Survival Data"
Chapman and Hall, 1984

Dahl O.J., Djikstra E.W. and Hoare C.A.R.
"Structured Programming"
Academic Press, 1972

Davidson R.N.
"Crime and Environment"
Croom Helm, New York, 1981

De Finetti B.
"Foresight: Its Logical Laws, Its Subjective Sources"
Kyberg and Smokler, ed., Wiley, New York, 1963

Department Of The Environment
"Handling Geographic Information"
Report of The Committee Of Enqiry chaired by Lord Chorley
Her Majesty's Stationary Office, London, 1987

Duncan R.
"Advanced MS-DOS Programming"
Microsoft Press, 1986

Epanechnikov V.A.
"Nonparametric Estimation Of A Multidimensional Probability Density"
The Theory Of Probability and Its Applications 1969, Vol. 14,
pp153-158

Evans D.J.
"Geographical Perspectives Of Juvenile Delinquency"
Gower, Farnborough, Hants., 1980

Evans D.J. and Herbert D. (Eds)
"The Geography Of Crime"
Routledge, London, 1989

Fildes R.
"An Evaluation Of Bayesian Forecasting"
Journal Of Forecasting 1983 Vol 2, pp137-150

Fildes R.
"Bayesian Forecasting"
in The Forecasting Accuracy Of Major Time Series Methods
Makridakis S., Andersen A., Carbone R., Fildes R., Hibon M.,
Lewandowski R., Newton J., Parzen E., and Winkler R.,
Wiley, New York, 1984

Fildes R. and Stevens C.F.
"Look - No Data: Bayesian Forecasting and The Effect Of Prior Knowledge"
in Fildes R. and Wood D. (ed) Forecasting and Planning,
Farnborough, Hants, 1978

Geary R.C.
"The Contiguity Ratio and Statistical Mapping"
The Incorporated Statistician, 1954, Vol. 5, pp115-145

Glass G.V., Willson V.L. and Gottman J.M.
"Design and Analysis of Time Series Experiments"
Colorado Associated University Press, Colorado, 1975

Glick B.J.
"Tests For Space-Time Clustering Used In Cancer Research"
Geographical Analysis 1979, Vol. 11 pp202-208

Hammersley J.M. and Handscomb D.C.
"Monte Carlo Methods"
Methuen, London, 1964

Hagget P., Cliff A.D. and Frey A.
"Locational Analysis In Human Geography"
Arnold, London, 1977

Harrison P.J. and Stevens C.F.
"A Bayesian Approach To Short-Term Forecasting"
Operations Research Quarterly 1971 Vol 22, pp341-362

Harrison P.J. and Stevens C.F.
"Bayesian Forecasting"
Journal Of The Royal Statistical Society (B) 1976 Vol 38, pp205-247

Hartigan J.A.
"Bayes Theory"
Springer-Verlag, New York, 1983

Herbert D.T.
"The Geography of Urban Crime"
Longman, New York, 1982

Home Office
"Criminal Statistics In England And Wales (annual)"
Her Majesty's Stationary Office, London

Home Office
"Criminal Statistics In England And Wales (annual)
Supplimentary Tables: Volume 3, (Tables By Police Force Area and Some Court Areas"
Her Majesty's Stationary Office, London

Hooper P.M. and Hewings G.J.P.
"Some Properties Of Space-Time Processes"
Geographical Analysis 1981, Vol. 13, pp203-223

Hope A.C.A.
"A Simplified Monte Carlo Significance Test Procedure"
Journal Of The Royal Statistical Society (B) 1968, Vol 30. pp588-592

James E.B.
"The User Interface: How We May Compute"
Computing Skills and The User Interface
Coombes M.J. and Ally J.L. (Eds)
Academic Press, London, 1981

Jeffrey C.R.
"Crime Prevention Through Environmental Design"
Sage Publications, Beverley Hills, 1977

Johnston F.R. and Harrison P.J.
"An Application Of Forecasting In The Alcoholic Drinks Industry"
Journal Of The Operations Research Society 1980 Vol 31, pp699-709

Johnson R.A. and Wichern D.W.
"Applied Multivariate Statistical Analysis"
Prentice Hall, New Jersey, 1988

Johnson R.J.
"Multivariate Statistical Analysis In Geography: A Primer On The General Linear Model"
Longman, New York, 1978

Kalbfleisch and Prentice
"Statistical Analysis of Failure Time data"
Wiley, New York, 1980

Kaplan E.L. and Meier P.
"Nonparametric Estimation from Incomplete Observation"
Journal of the American Statistical Association 1958, Vol. 53, pp457-481

Kendall M.G.
"Multivariate Analysis"
Griffin, London, 1980

Klauber M.R.
"Two-Sample Randomisation Tests For Space-Time Clustering"
Biometrics 1971, Vol. 27, pp129-142

Klauber M.R. and Mustachi P.
"Space-Time Clustering Of Childhood Leukaemia In San Francisco"
Cancer Research 1970, Vol. 30, pp1969-1973

Kleinbaum D.G. and Kupper L.L.
"Applied Regression Analysis and Other Multivariate Methods"
Duxbury Press, North Scituate, Mass., 1978

Knox G.
"Epidemiology of Childhood Leukaemia In Northumberland and Durham"
British Journal of Preventative Social Medicine 1964b, Vol. 18
pp17-24

Knox G.
"The Detection Of Space-Time Interactions"
Applied Statistics 1964a, Vol. 13, pp25-29

Knuth D.E.
"The Art Of Computer Programming Volume 2:
Seminumerical Algorithms"
Addison-Wesley, New York, 1971

Kolman B. and Trench W.F.
"Elementary Multivariable Calculus"
Academic Press, New York, 1971

Lawless J.F.
"Statistical Models and Methods for Lifetime Data"
Wiley, New York, 1982

Luger G.F. and Stubblefield W.A.
"Artificial Intelligence and The Design Of Expert Systems"
Benjamin/Cumming, California, 1989

Lo A.Y.
"On A Class Of Bayesian Nonparametric Estimates: I-Density
Estimates"
The Annals Of Statistics 1984, Vol 12., pp351-357

McCrea
"MICKA a FORTRAN IV iterative k-means cluster analysis program"
Behavioural Science 1971, Vol. 16, pp423-424.

McCulloch J.W., Smith N.J. and Batta I.D.
"A Comparative Study Of Crime Amongst Asians and Their Host
Population"
Probation Journal, 1974, pp1.

Mantel N.
"The Detection Of Disease Clustering and A Generalised Regression Approach"
Cancer Research 1967, pp209-220

Mardia K.V.
"Statistics Of Directional Data"
Journal Of The Royal Statistical Society (B) 1975, Vol. 37, pp349-393

Mawby R.J.
"Defensible Space: A Theoretical and Empirical Appraisal"
Urban Studies 1977, Vol. 14, pp169-179

Mayhew P.
"Crime as Opportunity"
Home Office Research Study No. 34, 1974

Muller H.G.
"Smooth Optimum Kernel Estimators Of Densities, Regression Curves and Modes"
Annals Of Statistics 1984, Vol. 12, pp766-774

Moran P.A.P.
"Notes On Continuous Stochastic Phenomena"
Biometrika, 1950, Vol. 37 pp17-23

Morris P.A.
"Combining Expert Judgements - A Bayesian Approach"
Management Science 1977 Vol 23, pp679-693

Morris P.A.
"Decision Analysis Expert Use"
Management Science 1974 Vol 20, pp1233-1241

Morrison W.D.
"The Interpretation Of Criminal Statistics"
Journal Of The Royal Statistical Society, 1897, Vol. 60, p1.

Newman T.G. and Odell P.L.
"The Generation Of Random Variates"
Griffin's Statistical Monographs, 29, 1971

Newman O.
"Defensible Space: Crime Prevention Through Urban Design"
MacMillan, New York, 1972

Openshaw S.
"The Modifiable Areal Unit Problem"
CATMOG 38, Geo-Abstracts, Norwich, 1984

Openshaw S. and Taylor P.J.
"A Million or So Correlation Coefficients: Three Experiments On The Modifiable Areal Unit Problem"
Statistical Methods In The Social Sciences,
N.Wrigley (Ed) Pion, London, 1979

Openshaw S. and Taylor P.J.
"The Modifiable Areal Unit Problem"
Quantitative Geography,
Wrigley N. and Bennet R.J.(Eds) Routledge, London, 1981

Pfeifer P.E. and Deutsch S.J.
"A STARIMA Model Building Procedure With Application to Description and Regional Processing"
I.B.G. Transactions 1980, Vol 5., pp330-349

Pike M.C. & Smith P.G.
"Disease Clustering: A Generalisation Of Knox's Approach To Space-Time Interactions"
Biometrics 1968 Vol. 24, pp541-546

Prospero Software
"ProFORTRAN 77 Uswer Manual (Version iid 1.2)"
Prospero, London, 1986

Pyle G.F.
"The Spatial Dynamics Of Crime"
University Of Chicago, Dept. Of geography Reserach Paper No 159, 197

Queen N.M.
"Methods Of Applied Mathematics"
Nelson, Walton-on-Thames, Surrey, 1980

Ripley B.D.
"Spatial Statistics"
Wiley, New York, 1979

Rhind D.
"A Census User's Handbook"
Methuen, London, 1983

Rosenblatt M.
"Remarks On Some Nonparametric Estimates Of A Density Function"
Annals Of Mathematical Statistics 1956, Vol 27., pp832-837

Reilly P.M.
"The Numerical Computation Of Prior Densities"
Journal Of The Royal Statistical Society (C) 1976 Vol 25, pp201-209

Savage R.E. and Habinek J.K.
"A Menu-Driven User Interface: Design and Evaluation Through Simulation"
Ablex, New Jersey, 1984

Schucany W.R. and Sommers J.P.
"Improvement Of Kernel Type Density Estimators"
Journal Of The American Statistical Association 1977, Vol. 72, pp420-423

Siegel S. and Castellan N.J.
"Nonparametric Statistics For The Behavioural Sciences"
Mc.Graw Hill, New York, 1988

Siemiatycki J.
"Mantel's Space-Time Clustering Statistic: Computing Higher Moments and a Comparison Of Various Data Transforms"
Journal Of Statistical Computing and Simulation, 1978, Vol. 7, pp 13-31

Silverman B.W.
"Density Estimation For Statistics And Data Analysis"
Chapman and Hall, New York, 1978a

Silverman B.W.
"Choosing The Window Width When Estimating A Density"
Biometrika 1978b, Vol 65, pp1-11

Sloman A..
"Artificial Intelligence Languages"
Intelligent, Knowledge-Based Systems, An Introduction
O'Shea T., Self J. and Thomas G. (eds) Harper and Row, U.S., 1987

Sparks R.F., Genn H., and Dodd D.
"Surveying Victims"
Wiley, London, 1977

Staniswalis J.G.
"Local Bandwidth Selection For Kernel Estimates"
Journal Of The American Statistical Association 1989, Vol. 84,
pp284-288

Tango T.
"Detection Of Disease Clustering In Time"
Biometrics 1984, Vol. 40, pp15-26

Tobler W.
"A Computer Movie Simulating Urban Growth In The Detroit Region"
Economic Geography, 1970, Vol. 46, pp 234-240

Wald A.
"Sequential Analysis"
Wiley, New York, 1947

Walker M.A.
"Some Problems In Interpreting Statistics Relating To Crime"
Journal Of The Royal Statistical Society (A) 1983, Vol. 146, pp281-293

Wegman E.J.
"Nonparametric Probability Density Estimation:1 A Summary
Of Available Methods"
Technometrics 1972, Vol. 14, pp533-546

Wilton R.
"Programmers Guide to PC and PS/2 Video Systems"
Microsoft Press, 1987

Winkler R.L. and Makridakis S.
"Averages Of Forecasts: Some Empirical Results"
Management Science, 1983, Vol. 29, pp983-996

Witten I.H.
"Communication With Microcomputers: An Introduction to the Technology
Of Man-Computer Communication"
Academic Press, London, 1981

Whittemore A.S., Friend N., Brown B.W. and Holly E.A.
"A Test To Detect Clusters Of Disease"
Biometrika 1987, Vol. 87 pp631-646

Wylie C.R. and Barrett L.C.
"Advanced Engineering Mathematics"
Mc.Graw-Hill, New york, 1982

Zellney A.
"An Introduction To Bayesian Estimation In Econometrics"
Wiley, New York, 1971

In depositing this copy of my thesis in the University Library I agree to the normal conditions of access (subject to the provisions on copyright and Library Regulation D.25).

Signed BRUNSDON, C F M

Date SEPTEMBER 1989

Department

Supervisors Name

Supervisors Dept.

THIS THESIS MAY NOT LEAVE THE UNIVERSITY LIBRARY NOR MAY IT LEAVE THE PREMISES OF ANY LIBRARY TO WHICH IT HAS BEEN SENT ON INTER-LIBRARY LOAN

Consultation Record

THESIS NUMBER

I recognise that the copyright of this thesis rests with the author and that no quotation from it or information derived from it may be published without the prior written consent of the author. I understand that it is my responsibility to ensure that no other person is allowed access to this thesis while it is entrusted to me.

Name in Block Capitals	Signature	Address/Institution/Department	Date
C. MCNEILL		ROYAL ULSTER CONSTABULARY CENTRAL FORCE LIBRARY	13/10/93.
		GARNERVILLE GARNERVILLE ROAD	
		BELFAST BT4 2NX	
CH GOODSHIP		DUNDEE UNIVERSITY	11/11/93.
HO LAW		room 262 Queen Anne's Gate HOME OFFICE	17/10/94
John McElhatton		Civil Eng	29/1/96
RICHARD KINGSTON	R.P. Kingston	TOWN + PLANNING	6/3/98
JONATHAN BROWN	J Brown	GEOGRAPHY/MATHS	17/5/00
SARAH THOMPSON	S J Thompson	GEOGRAPHY	11/01/01