Deriving and using an Object Based Model of a mapped area from a Feature Coded Representation.

Issue 2.5 dated 30th April 1986

Dr JMP Quinn, ICL Flagship Project, West Gorton, Manchester

Summary

This paper describes the techniques developed in deriving a spatial model of a mapped area derived from a feature coded digital map of that area. It describes how the model is used in conjunction with a database application (PLANES) which allows users, such as Public Utilities and Local Authorities, to set up and interrogate large quantities of property related information.

Introduction

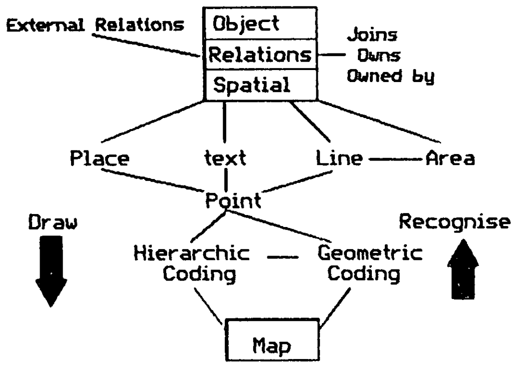
The work detailed in this paper forms part of a general investigation into the techniques of handling spatial data. A major impetus for this work came from the realization that a large amount of information is held in Ordnance Survey digital maps. The maps are often only used for simple drafting or providing a back drop for the production of drawings of service distribution networks. While these are sensible uses for the maps there is potential for much more and this paper describes the start of an exploration of such possibilities.

First, one must draw a distinction between a picture, or image of space, and the classification of space as a set of objects. A map is a familiar example of the former, whereas a Street directory or gazetteer is an example of the latter. A fundamental problem of spatial systems is that of linking these two representations to each other.

An image is normally represented by a rectangular grid of samples of a measured quantity, such as intensity, over an area. It can be converted into a line drawing by manual digitization or mechanised line trackers. This process introduces a varying degree of simplification in the image, depending on the type of image. The Ordnance Survey data is of this nature, and includes manually encoded information as to the meaning of a particular line (feature type) as well as textual data such as road and house names and house numbers.

The term object is used here to describe an entity that holds logical, textual and spatial information relevant to a specific item, such as a road or property.

The relationship between the set of spatial objects and its image is depicted in Fig.1. In this an object is shown to be composed of both logical and spatial information. The logical information relates an object to other objects while the spatial information defines the shape and position of the object in terms of primitive components. These are points, lines and areas. Any human readable information (text) also needs to be included. The components may also be defined in terms of sets of simpler components, e.g. an area in terms of its boundary lines, and this produces advantages in terms of consistency of detail, and reduced volume of data. In the final screen image everything is reduced to points of various colours. The act of drawing the map moves down from the objects through their spatial components. From there each element is reduced to a set of points in the displayed image. The act of recognition implies moving the other way. First grouping points into features then features into objects. With OS maps one is able to start the recognition process at the feature level, thus enabling a concentration on the process of object identification and handling.



In this experiment I explored the methods of transforming the feature coded image into an object based view. The techniques investigated are based on using the visual clues that enable a human to make sense of the patterns of coded lines that form the map. By employing these techniques it should therefore be possible to use the maps in a far wider arena than at present.

The main features of the Ordnance Survey data investigated are

a. The volume of data is large.
b. the data is often geometrically imprecise.
c. the spatial algorithms needed to extract meaning out of the data are highly dependent on the nature of the data.
d. the data appears to be coded in different styles with varying degrees of precision. In some cases the data has sections missing and is inconsistent with itself.
e. the cartographic data is provided in National Grid Coordinates i.e. two six digit numbers to be manipulated.

These considerations lead to an initial design philosphy for the system such that

i. the most useful key to large scale urban data is the road and street network, both logically and geographically. This later provided a convenient link into the PLANES system.
ii. all algorithms should allow a degree of tolerance to inaccuracy.
iii. while investigating an area the objects set up should be temporary, created as needed during a given session. Once the analysis appears complete it should be possible to save the objects in a data base from which they can be quickly restored when required. They can be modified when held in that database but the main aim is to improve the object extraction algorithms so that the amount of editing is minimal.

iv. major discontinuities in source data will be corrected by visual interpretation and manual intervention. The system should attempt to identify such situations and allow correction. Minor inaccuracies, such as lines that dont join exactly, can safely be interpreted and if appropriate corrected.

v. the means of interacting with the system should be simple, easy to use and extensible.

This approach places requirements on the supporting system such as

a. the system should provide a large store to accommodate the large data set

b. fast floating point manipulation because the data size is large and many geometrical calculations are involved.

c. Powerful multiprocess and multiuser facilities are also needed to facilitate independent and cooperative operation.

d. The software should be implemented in a high level language that provides flexible object handling, access to the system facilities and mixed language capabilities.

The Spatial Model

The spatial model is based on the road network of the area. It consists of sets of objects of various types. The objects are represented by data structures that include their logical, textual and spatial attributes. The main object types involved are defined for the Roads, Road Junctions and Properties.

The original data on which the model is based is provided by an Ordnance Survey digital map of Coventry at scales of 1:1250 and 1:2500 and accurate to 0.5 metres. The map covers an area of 72 square Kilometres in 110 map sheets. Metropolitan areas account for 84 500 square metres map sheets and country areas for 26 1 square Kilometre map sheets.

The data is held on a map sheet basis and consists of sets of over 300 feature types which can be lines (represented as lists of coordinates), text or point coordinates. These features are spatially not logically associated and each feature type is accompanied by qualifying information. In the case of lines and points the feature code and line style for drafting is included. For text the location and orientation is provided together with the text string itself.

The algorithms developed for extracting the objects from the feature coded map data can be viewed as a set of empirical rules which are specific to this style of feature coding. If the base representation is changed then the rules must be changed also. At the present time we have chosen to group the objects within map sheets, in the same way as the original feature coded data although this is not a fundamental obstacle.

Roads as Objects

The feature codes used for the road objects are those for the centre lines of the various catagories of road. The base data is thus the sets of centre line coordinates and associated categories together with the set of text strings that may include the name of a particular road. For each road a containing box is identified. The corners of the box are the minimum and maximum northings and eastings of the road coordinates. The boxes are used as a quick check for the proximity of objects.

A road is identified as a set of centre line vectors, of a specific road centre line feature code with a single associated name. There are 10 different feature codes associated with road centre lines.

The strategy for the allocation of names to roads is

a road is a set of vectors with a single name within 2 meters of the line of vectors and an orientation parallel to those vectors

or

a road is the set of centre line vectors has no name
and is either continuous with a named road whose name it can take
or joins or is joined by a named road whose name it can take

or

a road is the set of centre line vectors road with multiple names associated with it along its length
and can split up the original road at
either junctions with other roads*
or bends so that each section of the original road has a unique name.

* In the case where there are several junctions between two names then one is chosen by looking at house numbers and choosing the junction nearest the smallest number. This technique relies on the Ordnance Survey apparent convention of orientating house numbers with respect to the road in which they are located.

If none of these is possible then the fact is reported for manual intervention.

If was found that there were no roads in this area which had overlapping names but in that case two coincident roads would be defined.

After the first pass about 60% of the roads have been named uniquely. The final success rate is 90%.

Road Junctions as Objects

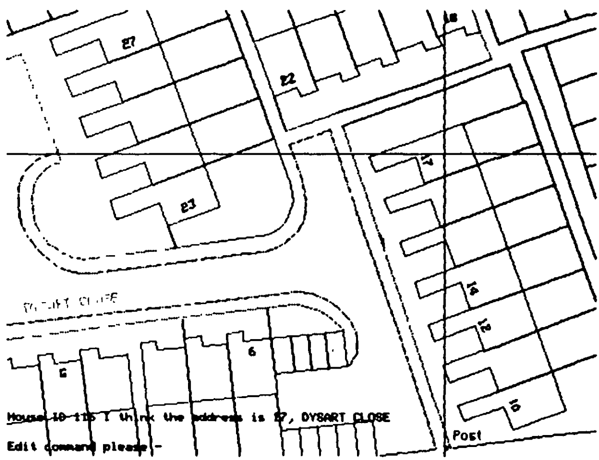
Part of the road naming algorithm above requires the road junctions to be determined. This is achieved by first checking whether two roads can intersect by using their containing box. If their boxes overlap then a geometrical algorithm is used to determine the actual point of intersection [Ref 1]. The junction object includes the road identifier, the intersecting road identifier the point of intersection, the type of junction and the number of intersections that share the same junction together with their identifiers.

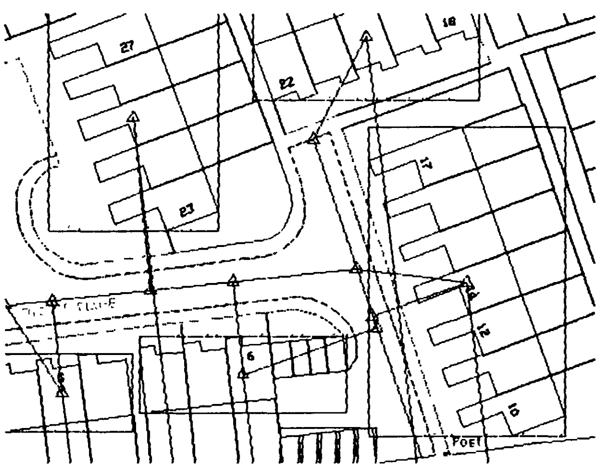
Properties as Objects

There are several feature codes that relate to property outlines. At this scale (1:1250) the shape of a building is indicated and in the case of terrace housing it can be very complicated. The data may be found in several sections not necessarily contiguously placed in the map datafile.

The strategy for identifying properties is based on the definition of a property as a closed polygon bounded by lines of one of the OS building feature codes (1..6). The area defined by the polygon will usually include either a number or a name text string. It will also has an associated road within 30 metres in a direction indicated by the orientation of the text.

The next two figures are direct screen dumps. Fig 2 shows the property polygons in red including the numbers. The dividing walls are shown in blue. The road edge is shown as a chained line in red and the green lines show other boundaries. The cursor is placed over the house of interest and the corresponding address is displayed as 17, Dysart Close.





The geometry involved in attaching the properties to the roads is shown in Fig 3. The containing boxes for the properties are shown in black. Also shown are the road centre lines and the construction lines dropped onto the nearest roads' as perpendiculars from the centre of the containing rectangle. If that is not possible then the nearest point on the road is taken. The road chosen is then the closest one in the direction of the numbers.

In properties, such as terraced housing, which include multiple numbers within the enclosing polygon the boundary/fence feature codes are used to split up the property polygon into house polygons each with its own associated number or name. Flats are identified conventionally by a name and the range of flat numbers given by a text string of the form A to B.

In many cases it is necessary to interpolate between the OS house number data in order to allocate missing numbers to the houses. However, the vagaries of the system of assigning numbers to properties precludes this in every case and the information must provided manually.

If any property crosses a map sheet boundary then the polygon is currently closed along the boundary.

The success rate of the above algorithm varies widely depending how well the text direction convention is adhered to and how consistent the digitiser was in keeping to closed polygons as outlines. In some areas the success rate is 80% in others it is closer to 50%.

DataBase structures

There are three main databases involved: the map database and the object database are used to hold the spatial data and the Planes database which holds other textual attribute data.

The spatial databases are based on block structured files. Position in the file is given by a File Address: the block number and offset in block at which the item starts.

Both the map and the object database are organised on a map sheet basis. They have an associated index file that gives the northing and easting coordinates for each map sheet as well as the map width in meters together with the file address of the start of its data. This allows the display of any part of the mapped area by selecting the map sheets which fall within the area of interest and then clipping the map sheet's data to the screen.

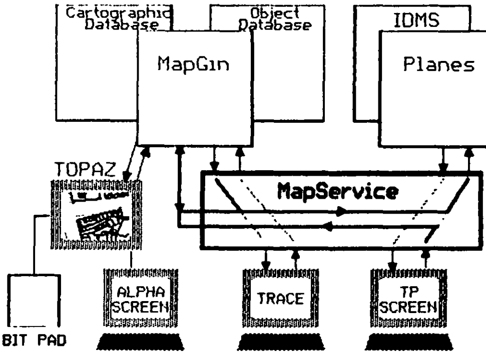
The data for each map sheet is prefaced by a feature index that indicates which codes are included and the file address at which they start. The line data itself is held as a sequence of coordinate pairs, relative to the map sheet base, with a count of the number of points at the start of each sequence.

The map database allows records which span over more than one data block, the object database does not. This is because the map data is not intended for modification locally but by updates from the Ordnance Survey. The Object data is intended to be modified by improving the object extraction algorithm when a weakness in its operation is detected and then overwriting the old data.

Experimental Spatial System

The main parts of the experimental system are shown in Figure ?. It consists of three main subsystems that can be run independently to provide a subset of the total facilities, or together so that they can cooperate to provide the total set.

The subsystems are : MapGin which handles all the Cartographic data, PLANES which handles the main property attributes and MapService which intercepts messages from either PLANES or MapGin and routes them to the appropriate recipient dependent on the context.



The lines show the interconnections and routes used. Each of the subsystems has its own terminal for trace information and MapGin also has a graphics terminal, labelled Topaz with its own alphanumeric terminal and bitpad for cursor control.

There can be more than one MapGin subsystem connected. Each can either interact separately with PLANES or act as a slave to another MapGin sub system. Both PLANES and MapGin can operate independently and need only be connected through MapService when either wants data from the other.

The system described runs on an ICL mainframe which supports the operating system VME, (Virtual Machine Environment) capable of supporting multiple users. A software environment has been developed to provide all the necessary graphical and communications facilities required. The basic graphical facilities are device independent and can be mapped on to various graphical devices in order to display and interact with the maps. Such devices are currently connected asynchronously.

The set of devices that are supported are the Tektonix colour terminals 4501, 4701 and 4695 (printer), the Topaz greyscale graphics terminal, ICL 4602G terminal, ... It is simple to add further device drives.

MapGin

The user is able to interact with MapGin through a simple command language. Any positional information required is provided by using a cursor. The type of interaction is controlled by MapGin.

The user can select an area of interest and may move interactively about the mapped area, changing the scale as desired. It is also possible to move using the text feature included in the map data as a gazetteer. All positional information is held and presented in National Grid Coordinates.

In a given mapsheet the roads and junction objects can be determined by using the command WALK and the property objects by the use of the command PROPERTY. Once these have been used the objects can be saved using the SAVE OBJECTS command or interrogated using the STREET command for the roads or the

ADDRESS command for the properties. In the latter command the graphics cursor is used to point to the road or property of interest.

A typical sequence of commands with explanation in {} would be as follows

map 39	{ display map 39, a 500m sq map at coórdinates 434500, 279500 }
walk	{ identifies roads and junctions }
property	{ identifies all the properties and allocate to roads }
save 39	{ writes objects to a database }
mz	{ zooms into a particular area indicated using the cursor }
address	{ point to a building and display its address }
street	{ point to a street to determine its name }
...	{ continue with the interaction }

PLANES

PLANES is an ICL application currently in use with a number of customers. It is an IDMS based Transaction Processing (TP) Service which provides access for a number of different applications grouped round an address directory handler. The directory handler maps an address to a structured key that can be used to retrieve the corresponding data from the supporting service. The directory handler also determines the hierachical structure of the data.

In this experiment a district of Coventry is split up into roads and the roads into properties and sub-properties. There is no concept of the next road in the data, there is, however, the concept of next house in a road, on the basis of its number in the road. Properties can be also be grouped together into sets on the basis of common items of data, irrespective of their actual address. Such sets are given a specific code for access called an entry point.

The application used in this experiment was the PLANES System Interface Manager PISM. This allows the user to set up and interrogate data, keyed on address, under three headings: Planning, Commercial and General Enquiries. Once the heading is chosen it remains in force until specifically changed.

The usual way in which the user interracts with PLANES is by a set of formatted screens. The user inputs his enquiry by just completing the appropriate fields on the screen and PLANES replies in a like manner, one screenful at a time. A user thus can move through a particular set of data by inputting a walk to next item command.

Each screen of information has its own identifier. When MapGin supplies the input data instead of a user the screen code must be sent with the message to mimic the action of the TP terminal. For example to initiate an enquiry on an address such as 1, DYSART CLOSE the message would be sent as

```
PI00+1+                       +DYSART  CLOSE        +
```

where + represents the field separation character. On receipt of this address PLANES will retrieve the data and display it on a new TP screen appropriate to the enquiry.

MapService

This is a special purpose program that uses VME communications facilities to connect the various components of the system together. It intercepts all the messages between PLANES and its TP terminal and can reroute all or a specific set of messages to MapGin. It can support the broadcast of messages to all the MapGin processes connected or handle them individually.

Using the Spatial Model

The interaction between PLANES and MapGin can be initiated from either end. Examples of such interactions are as follows..

a. MapGin can be used to display the area of interest and once the analysis has been carried out road names and property addresses retrieved by just pointing at the displayed image of the road or property. When connected to PLANES the address generated can be used to extract and display any associated attribute data. For example property descriptions, owners names and occupations.

b. The PLANES user may browse through a set of properties one by one and for any one can input the command VIEW. This command is intercepted by MapService and routed to MapGin so that the property can be displayed on the screen. In order to do that MapGin causes the PLANES screen to be redisplayed and a copy sent to MapGin. From this the position of the property can be determined and it is displayed at a scale that shows an area of 100 meters square centred on the property.

c. The PLANES user can select a property in a road and then WALK from one property set to another. At any stage he can request a walk to the next road, left, right or straight on. This command is routed to MapGin in order to determine the current road and property. The next road is selected on the basis of the nearest junction and then the first house chosen that satisfies the direction (left, right or straight on). This address is then passed to PLANES so that the enquiry can continue from the new property.

d. It is possible to use MapGin to determine and display the distribution of attribute data stored by PLANES within a given area. It is first necessary to select the shape of the area to be examined. This may be a circle, a rectangle or a irregular polygon. Commands are then sent by MapGin to PLANES that cause it to walk through the set of properties with that attribute data and route the resultant PLANES data to MapGin. MapGin then checks whether the property is within the area. If it is the property is highlighted otherwise it is discarded.

Conclusions

The MapGin experimental system has demonstrated that it is possible to construct an object based model of a feature coded area with an acceptable deqree of success. It is then possible to use that model to provide a graphical front end to an existing object based enquiry system such as PLANES based solely on the message passing employed, to communicate with the user, such that the two parts of the system complement each other and the whole is greater than the sum of the parts.

Acknowledgements

I would like to thank Dr I Moshkun of ICL whose help and encouragement made the goinq easier, Mr N Perry of ICL Knowledge Base Systems Project who's initial work laid the foundations for this experiment, Mr JM Pratt of ECRC, Munich, for his help during the early stages of the project, Mr CJ Skelton for his financial support and understanding throughout the project, and to the members of the PLANES team for funding part of the experiments and mending their part when broken and others colleagues too numerous to mention...

References

A Programmers Geometry
A Bowyer (1983)
Pub: Butterworths.

Fundamentals of Interactive Computer Graphics
JD Foley and A Van Damm (1982)
Pub: Addison-Wesley.