VECTOR – RASTER CONVERSION

Tran Quoc HUNG

Department of Photogrammetry
Technical University of Budapest
H–1521 Budapest, Hungary
E–mail: hung@epito.bme.hu

Received: May 15, 1995

Abstract

This paper examines the characteristics of vector and raster databases and compares the application of these databases in geographical information systems. This paper shows in particular a method of conversion of vectorial database to raster and the application of this method to create land cover database of Hungary.

Keywords: GIS system, raster–vector conversion, landcover database.

Nowadays the geographical information system (GIS) helps to solve a wide range of complex problems. With the rapid development of computers we have the possibility to process and visualize not only vector information but also images and raster form data. We can imagine a raster file as a data matrix, the elements of which reproduce the surface as it is in reality. The raster form storage helps the mathematical processing of the database (e. g.: points, local, geometrical, etc.). In the case of some works (e. g.: area analysis, environment protection, etc.) that demand not only arithmetic but also visualizing the raster form is clearer and more useful than the vectorial. The most GIS system endeavours to handle the vectorial and the raster databases in the same time (hybrid raster – vector editing). It utilizes all advantages of both the databases, so the achievement and the efficiency of GIS system increase greatly [PIWOWAR, 1990]. There are several ways to create a raster file, one practical solution is to convert the exiting vector database into raster in accordance with the actual task. In the following a possible solution of the vector – raster conversion will be presented.

1. Characteristics of Vector Database

We can create the vector database in several ways (e. g.: digitizing, scanning, deduction from a model, etc.). The element of the vector database can be different depending on the definition of the system (e. g.: point, line polygon, text, etc.). The storing of element happens as follows:

- Description of element (type, layer, colour, thickness, etc.)
- Connection between an element and the next one (a pointer may show the place of the next element in the file)
- The coordinates of the element (X, Y or Z in accordance with the actual model)

As the vector database consists of the coordinates of the elements it is complicated to work with a surface created from more than one element and it is not capable of processing the elements of the surface. To define the surface it is necessary to carry out more examinations one after the other. This method is based on a model which is called topological data model. Topology is spatial relationship between connecting or adjacent coverage features. For example, the topology of a surface includes its nodes and its left and right polygons. By storing information about the location of feature relative to other features, topology provides the basis for many types of geographical analysis without having to access the absolute locations held in the coordinate files (e. g.: connectivity, route finding and continuity are all derived through topology). This data model is applied in the most GIS systems (e. g.: ArcInfo, Integraph MGE, etc.).

The raster database is deducted from the vector database with the necessary resolution depending on the character of the task. The vector database must contain the coordinates of the objects that serve in the conversion for the deduction of raster elements. The type of element depends on the grouping of objects, the three most common ones are the following:

- point elements with the matching symbols from which we get to know the position of the point on the map and its type (it can be a line element for the stating the final point of which is the same)
- line elements and borders of the surface (line or polygon element)
- text elements that show the character of the surface (surface code)

To avoid the problems of conversion the vector database (input base) should meet the following requirements:

- every object should be situated on different layers because this defines the type of the object.
- the surface should always be closed — that is the sides of the surface should be its borders at the same time.
- the place of the surface code should be inside the borders of the surface.
- the rank of the surface code should be defined unambiguously.

2. The Characteristic of Raster Database

The raster file can be imagined as a data matrix. The distance between the rows and columns of the matrix depends on the resolution of the raster. The employment of the database, the scale of the map and the efficiency of the computer have a great influence on the choice of the resolution. The number of the objects on a given territory defines the number of bits characterising the matrix point (e. g.: if the number of the objects is 14 then an element occupies 2 bytes or 16 bits in the file). The bit storing a characteristic of the surface has the value of 1 while the others have 0. A raster element can store more pieces of information because for example the electric lines and the surface elements underneath the electric lines seem to be the same but in the raster file they appear as two different elements, so both bits belonging to the object have a value of 1, the others 0. This kind of storing allows us to define the type of the object with the examination of the bits or by masking the bit to process the database with mathematical methods.

3. The Process of Vector – Raster Generation

The process of vector – raster generation can be divided into the following parts:

- reading of database
- generation of lines elements and borderlines (line generation algorithm)
- organizing of surface codes
- generation of surface elements (surface generation algorithm)
- writing of raster database

Reading the data means to read element describing parts, element coordinates and eliminating redundant information. With the help of the layer and the type (point, line polygon or text) of the element we can define the character of lines and surface codes. These data get to the memory of the computer into blocks occupied by the program.

With the help of the line generation algorithm we convert the line (defined by its two final coordinates) into raster elements. Several algorithms have been worked out for this purpose, we present only one possible solution. Starting from the first point of the line the raster elements follow each other both horizontally and vertically (dx and dy) depending on the steepness of the line and the last raster element is in the final point. (*Fig. 1*). The line can define an object or a borderline, it is the last bit which differentiates (if 0: object else borderline). In the case of a point

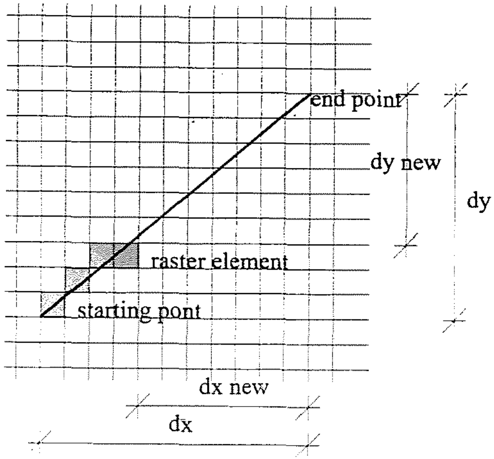

Fig. 1.

element there is only one raster element, because the starting and the final points are identical, so dx and dy are 0 [ACKLAND, 1981]. During the vector – raster conversion the line algorithm is very important because not generating precisely it can cause a change in the size of the given territory (in case of a borderline) or inaccuracy of the data (in case of a line).

As after the generation of a line the borderlines of the surface are not differentiated only the rank of the surface code helps us to decide which borderline belongs to a certain surface. This way the subordination of different types of surface is necessary. The arrangement of surface codes is necessary because the surface generation algorithm is happening from the higher rank code to the lower ones. The value of the raster element, which is on the borderline, is always identical with the higher rank surface raster value (e. g.: if a forest and a lake in that forest have a common border then borderline belongs to the forest because the forest is on a higher level than the lake). Here we have to be careful: if the distance between the borderlines belonging to a surface is smaller than the raster resolution, it is advisable to order more surface codes to it, otherwise the programme considers it as a closed surface.

The surface generation algorithm starts from a point inside a surface and examining a certain direction it orders the suitable value to the other points. This process continues until reaching the borderlines of the surface.

The basic principle of possible surface generation algorithm can be seen in *Fig. 2.*

The program first gives raster value to the element of the row containing the surface code, in the meantime it examines the upper and lower

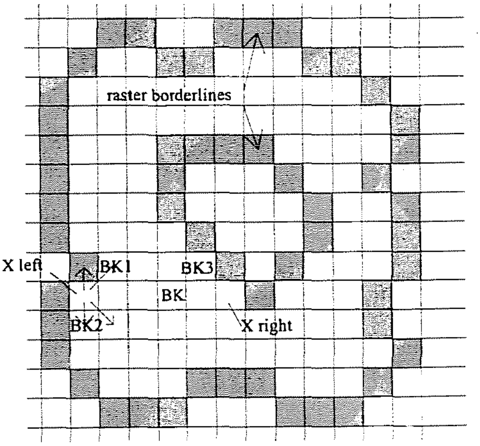

Fig. 2.

rows. If during the examination it finds the borderline the program gives the surface value to the element of the borderline and orders a new surface code to the examined row. Every new surface code gets a code index on the base of which the filling of the other rows is taking place. As you can see it in the figure the original code is marked with BK. The filling of the row is taking place in both directions (right and left). The examination of the rows up and down starts from Xleft and continues to Xright. Arrows show the place of the examination points. BK1, BK2, etc. are new surface codes created during the examination [PAULIDIS, 1979].

Beside the database the raster file can contain the number of the map shifts, the number of rows, columns and references that show where and what kind of further information can be got. These data make the access to the information for processing easier.

4. Applying Vector – Raster Generation Algorithm in Order to Create a Hungary Landcover Database

With the growing demand and technical level the GIS system plays a more and more important role in the field of telecommunication. Without taking the surface of the earth into consideration the expansion of the telecommunication network is not possible. In the planning of these networks it is very important to examine the reflection of waves influenced by the relief of the surface and by the natural and artificial objects situated on the territory. We can get information about terrain and objects on the surface from the DTM and the landcover database. The Hungary landcover database was realized by Photogrametry Department and Geomatik Ltd. with the Frequency Management Institute of Hungary cooperation.

The natural and the artificial terrain object's height and character determine the reflection of land surface. In this project 14 different covering types were defined. The determination of the different elements was made by topographical maps, aerial photos and field control. The different covering types were:

Point elements:

- chimney,
- antenna,
- tower, etc.

Line elements:

- river,
- road,
- railway,
- aerial electric line (high voltage),
- bridges/overpasses.

Surface elements:

- meadow, low vegetation,
- water surface,
- bushy district, medium high vegetation (max. 4 m),
- low forest (max. 15 m),
- high forest (greater than 15 m),
- low building, villages (max. 8 m)
- medium building, small town (max. 15 m),
- high building, cities (greater than 15 m).

The creation of the database was made on a Microstation graphical system with PC-s and Intergraph workstations. The database consists of two parts: the vector and the raster databases.

The vector database has been put into the computer with the digitalization of 1 : 50,000 topographical maps. The elements of objects are represented with different colours and thickness on different layers so by switching the layers on and off there is a possibility of clear representation. *Fig. 3* shows a vector database.

The raster database was grouped according to the 1 : 100,000 scale EOTR map with a resolution 50×50 and 200×200 m. The length of a 1 : 100,000 EOTR map shift is 48,000 m, the width is 32,000 m so a row of a raster file contains 960 or 480; a column contains 640 or 420 elements depending on the resolution of 50 or 200 m.

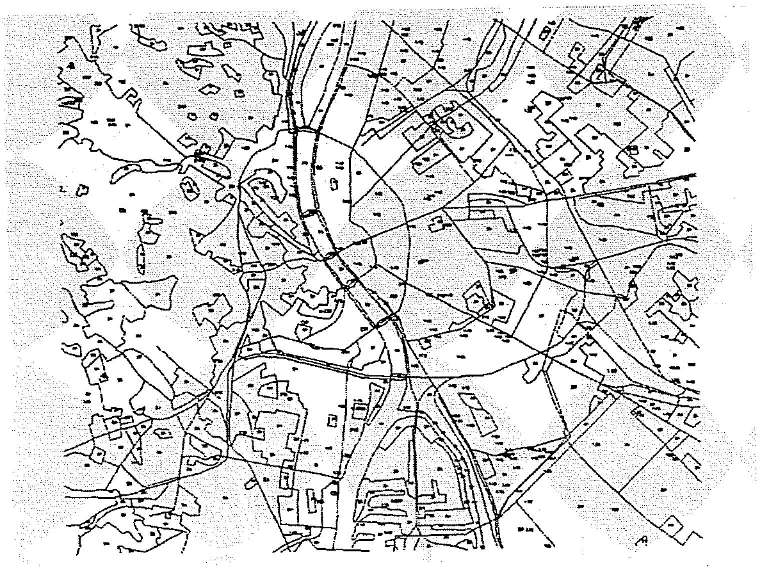

Fig. 3.

The time of conversion is determined by the number of rows and columns of raster file and by the efficiency of the computer. Because of the limits of PC-s the conversion took place on workstation (UNIX operation system).

Fig. 4 shows a raster database with a resolution of 50 m.

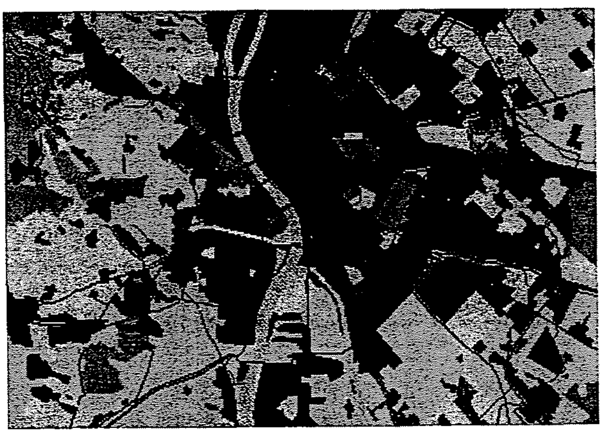

Fig. 4.

References

ACKLAND, B. D. – WESTE, N. H., (1981): The Edge Flag Algorithm a Fill Method for Raster Scan Display. *I.E.E.E. Transactions on Computer,* Vol. 30, p. 41.

PIWOWAR, J. M. – LEDREW, E. F. – DUDYCHAX, D. J., (1990): Integration of Spation Data in Vector and Raster Formats in Geographical Information System Environment. *International Journal of Geographical Information System.*

PAULIDIS, T. (1979): Filling Algorithms for Raster Graphics. *Computer Graphics and Imager Processing,* Vol. 10, p. 126.

INDEX