

# SEMI AUTOMATIC DIGITIZING OF CONTOURS FROM 1:25000 SCALED MAPS

C. Helvacı<sup>a\*</sup>, B. Bayram<sup>a</sup>

<sup>a</sup> YTU, Civil Engineering Faculty, 34349 Yıldız, Besiktas Istanbul, Turkey – (chelvacı, bayram)@yildiz.edu.tr

Commission III, WG III/4

**KEY WORDS:** Digitization, Extraction, Rectification, Algorithms, Raster, Vector

## ABSTRACT

It is an expensive and time consuming task to digitize 1:25000 scaled maps which are made in the past with analogue methods. The development in digital image processing depending on the development of hardware technology made this process less expensive and faster. This study is based on this reality. In this thesis a new algorithm which aims to decrease human work time by making the work carried out to digitize 1:25000 scaled maps easily and less time consuming was developed. The first step of the process is interactively rectification of the map. The grid network is used for rectification. Scanned maps at the scale of 1:25000 and the training area on this map which is given by the operator that is showing the elevation contour pixels are used as input data. The other elevation contour pixels are determined automatically. Elevation contours in vector format are created from the elevation contours in raster format. Produced vectors may contain errors depended on the topographic condition of the map area. Elevation contours with XY coordinates are generated after the manual editing. The Z values of these contours are given by the operator before the end of the process. Finally, produced data is saved in ASCII format which is supported by common CAD, GIS and engineering software. The output file contains X, Y, Z values delimited with comma (“,”).

## 1. INTRODUCTION

Automatic information extraction from map images is not a trivial task since objects in the charts must be interpreted from their spatial characteristics such as shape, scale and contours (Esposito F., et al.1998). Vectorization, also known as raster to vector conversion, is a process that finds the vectors – straight line segments – from the raster images (Wenyin, L. Et al.,1999). The contour-pixels detection consists of edge detection, thinning, and linking operations. These are window operations, in which the output at a pixel is based on the value of the input pixel and the value of its neighboring pixels (Chung Y, et al.,1995).

In Turkey many of the 1: 25 000 scaled maps are analogue. Geodetic and planning applications need these maps in digital format. Conventional digitizing process of 1: 25 000 scaled maps according to the topography of the area takes a very long time; sometimes a month. When this process is considered for each of the governmental and public institutions, the abundance of cost and time spent can be imagined. On the other hand, accelerated production of digital terrain models gives a possibility to design the projects in minimal time and cost. Algorithms like these, will avoid the errors that occur because of the human factor.

The aim of this study is to minimize the digitizing time of 1: 25 000 scaled maps. From the results of the study, it can be declared that this process reduced the digitizing time to a period between 4 hours and a week. Naturally, this time depends on the topological structure of the area and deformation of the map.

This software can be used with 200 dpi resolution scanned maps.

## 2. THE SEMI-AUTOMATIC DIGITIZING ALGORITHM OF CONTOURS

Semi - automatically recognition of the contours in 1:25000 scaled scanned maps is made by the process below:

### 2.1 Rectification

2-D Affine transformation method is used to rectify the 1: 25 000 scaled map.

### 2.2. Collecting the Training Samples

The method works with 24 bit true color image. To collect training sample contours, the points below should be checked:

1. The contours that have the same color but different gray values should be grouped.
2. The training samples should be in the middle axis of the contours.
3. The contours which have the same color but different brightness should be grouped.

To have accurate measurement results, the collected samples have to be distributed normally. There is no need to make any statistical adaptation tests to control the distribution of pixels selected by the user. Instead, the algorithm in section 2.3.1. is able to demonstrate the contours in whole image depending on the training measurement of users. With this method, the most accurate group is chosen interactively.

### 2.3. The Automatic Recognition of the Contours

The linear approximation consist of contour tracing and approximation operations (Chung Y, et al.,1995) . Algorithm of deformable model method needs premise measurements for

---

\* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

starting parameter and these measurements are given to the algorithm by the operator. (Vosselman, 1998). These kind of algorithms have snake algorithm basis. Snake algorithm uses the energy function and aims at the result product of the two methods mentioned above.

This method has some difficulties:

- i. Relating to the radiometric quality of the image, the alteration threshold of pixels in gray values are not constant.
- ii. Because of the noise in the image, some of the road-making pixels have different gray values and this causes broken contours.
- iii. In urban areas, objects on the roads may cause the image of the road to appear broken.

The Algorithm developed in this study is principally a guide algorithm. However, it uses the gray values belonging to the contours that the operator chooses as the starting parameter.

### 2.3.1. Determining the Colour Threshold of Contours:

This Process is used to define raster to vector conversion parameters. The parameters are calculated by training contour samples. Except for this, the different ground colors on the map effects the contour pixels color directly. Thus, with this step, a new training group is built for each ground color and different threshold values are calculated for each group.

The collected data is the gray value of the pixels belonging to their RGB bands. For calculation process, minimum and maximum RGB values are found for each group.

$$\begin{aligned} R() &\in Map() \\ G() &\in Map() \\ B() &\in Map() \end{aligned} \quad (1)$$

$$\begin{aligned} R_{max} &\in R(), R_{min} \in R() \\ G_{max} &\in G(), G_{min} \in G() \\ B_{max} &\in B(), B_{min} \in B() \end{aligned} \quad (2)$$

where  $R(), G(), B() =$  bands of scanned map image.  
 $R_{max}, G_{max}, B_{max} =$  maximum gray values of the contour sample set.  
 $R_{min}, G_{min}, B_{min} =$  minimum gray values of the contour sample set.

**2.3.2. Raster to Vector Conversion:** To start the automatic elevation value election process, the image matrix is scanned from left to right. If the scanned pixel and at least one of its eight neighbors are in the gray value of contours, they are chosen as the starting pixels. This  $x, y$  image coordinated pixel is taken as the  $i, j$  coordinated central element of  $3 \times 3$  nucleus matrix. If the gray values of pixels in  $R_{min} \sim R_{max}, G_{min} \sim G_{max}$  and  $B_{min} \sim B_{max}$  (Eq. 2) then it is given a value of 1 and the others 0. The center of gravity is calculated using the coordinates of pixels that have 1 gray value. The binary pixels are created like Costa (2001). This value is taken as a central pixel of the new

nucleus matrix. If the centers of gravity of binary image pixels intersect with the nucleus matrix central pixel, the closest 1 gray value pixel is chosen as the next target.

Thus, the first matched pixel is used as the starting point of the polygon. The recent pixel coordinate is added to the polygon. Again the recent pixel is taken as the center pixel of nucleus matrix. For these steps, gray values of pixels which are in the elevation value threshold are taken as 1 and the others are taken as 0. While calculating their center of gravity, the central pixel, the neighbors of the former central pixel and nucleus matrix are not taken into consideration. The process is repeated in this way.

**2.3.3. The Node Elimination Process of the Vectors:** After the vector data production process, approximately 500 000 to 2 000 000 broken contours occur, according to the topological structure of the map. This abundance of broken contours requires a large memory and increases the time spent for the analyses. The increase in the analyses time is geometrically proportional with the number of broken contours.

Line generalization algorithms are not used for decreasing the number of broken contours, because the broken points mentioned are in the middle of the pixels and they make an angle of 45 or multiples of this number. These kind of sharp angles between broken contours cause the line generalization algorithms to increase the number of broken points.

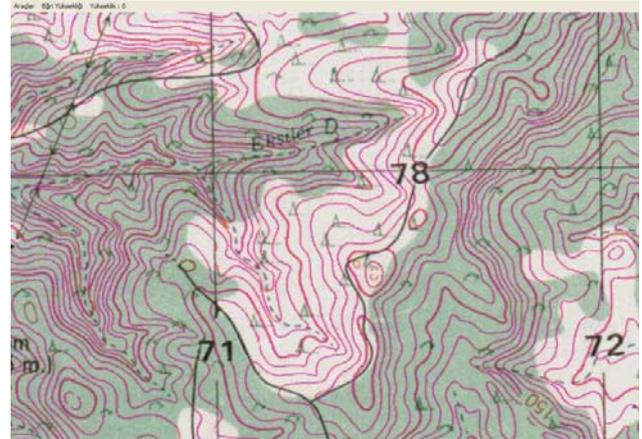


Figure 1. Vectors from raster (pink colored)

In this study, a method is chosen to delete the broken contours systematically. The only parameter of the method is the number of the nodes to be deleted. This choice is made by the user interactively, according to the topological structure of the map (Figure 1).

**2.3.4. Editing the Broken Contours:** Elevation lines can be broken because of some deformations or other cartographical objects on the map. The greatest factor that affects the speed of this step is the number of broken contours. Another important parameter is the threshold value distance for connections to be chosen, since the elevation lines are very close especially in hilly areas. If a wide threshold value is chosen, neighbor curves can be joined. So, it is concluded in this study that this choice should be made by the operator.

#### 2.4. Deleting the Vectors According to the Length

This process is designed for saving time. While editing the map, the map and the contours have to be superimposed. To realize this process the CPU needs a considerable time, for example: an Intel Pentium 4 2.4 Ghz processor with 512 MB RAM can run this process for 2 million nodes in 3 seconds. This time delay results in loss of concentration for the user. No graphic accelerates (DirectDraw, Direct3D, OpenGL, etc.) have been used for this step of software. But all the imaging process is done by GDI.

For this step, the user defined contours under a predefined length (in pixel value) are automatically deleted. On the other hand, the small cartographic objects which have the same threshold range with training samples are deleted automatically too. Then, the editing time and user mistakes are decreased.

#### 2.5. Editing the User Marked Contours

Due to the scanning or deformation of the map, broken or wrong contours can occur in the elevation lines after automatic elevation line recognition process.

The steps below are practiced to correct these mistaken lines:

- Deleting the vectors that occur while digitizing the cartographic objects that were not elevation lines as elevation lines
- Drawing the lines that were not drawn or that were deleted because of wrong formation
- Changing the places of broken points of existing elevation lines
- Deleting the broken points of existing elevation lines for generalization
- Separating the wrongly joined elevation lines
- Joining the broken elevation lines

#### 2.6. Determining the Elevation Values for Edited Contours

The user chooses the elevation lines one by one and gives their values. This step can be considered not to be a lengthy process. When the broken lines are joined, the number of new lines is very few. For this reason, this step takes 30 minutes to 1 hour according to the capability of the user.

A satisfactory result can only be obtained by giving Z value to the lines where the inclination changes. In the developed software, there is a command to delete the lines which are not given elevation values. By this method, the rest of the lines can be deleted at once.

### 3. SOFTWARE OUTPUTS

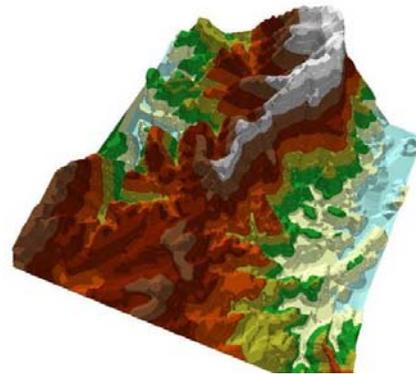


Figure 2. 3D elevation model of results (Model Made with © ESRI ArcInfo)

Broken contours are separated according to their XYZ values in ASCII file method, which is a method supported by most of the CAD, GIS and other engineering software systems (Figure 2). Thus, the contours can be saved as xyz extension files. This method makes it possible to get output from overloaded files on the hard disk in accommodation with every program.

### 4. CONCLUSION

The developed algorithm aimed to accelerate the digitizing process of 1:25 000 scaled maps semi automatically; independent from the map's topological structure. The study has achieved this aim.

Another factor that affects the method of the study is the resolution of scanned image. This algorithm is developed considering the thickness of the contours as 1 or 2 pixels. Another fact that the image quality is that, the image should be scanned in RGB. Otherwise, it is difficult for the algorithm to differentiate the contours from other cartographic objects. This difficulty increases the time spent for designing the lines, which prevents the study from reaching its goal.

The most important difference of this algorithm is its contours-oriented method. The main subject of this study is recognition of the contours from their center points. The 1:25 000 scaled maps have contours of 10 m. interval. So as not to reduce the elevation sensitivity, the contours are recognized from their middle points. However, many vector algorithms realize this process by converting the borders of the pixels that form the lines to line pieces. This method increases the time spent for designing the contours.

In this algorithm, the time spent for semi-automatic recognizing of plain and rough areas on a map is about 4 hours to 1 day. For extremely rough areas, this process takes 1 week to 10 days. With classical methods, this step takes 3 days to 1 week for plain and rough areas and 1 week to 1 month for extremely rough areas.

The developed algorithm works with a large amount of data. The information gained from this data can be much more complex, according to the roughness of the area or the starting parameter that the user chooses. Therefore, this software has been developed in C++ language. Since the C++ language is a

low-level language, the file size of the program is small. Strong memory management and speed is a “sine qua non” for an image processing operation, so that it was inevitable to choose the C++ language for this study.

The reason why the result file is saved in XYZ ASCII is that, it is supported by CAD, GIS and other engineering software. Thus, the data results and digital elevation models can be formed.

## 5. REFERENCES

Chung Y., Prasanna V. K., Wang C., 1995. A Fast Asynchronous Algorithm for Linear Feature Extraction on IBM SP-2. Workshop on Computer Architecture for Machine Perception, September, pp. 294 - 301.

Costa R. D., John Starkey, 2001. Photolin: A Program To Identify And Analyze Linear Structures In Aerial Photographs, Satellite Images And Maps. Computers & Geosciences 27, pp. 527 – 534.

Esposito F., Lanza A., Malerba D., Semeraro G., 1998. Information Capture from Topographic Maps Using Machine Learning. Proceedings of Joint Workshop of the Italian Association for Artificial Intelligence (AI\*IA) and the International Association for Pattern Recognition - Italian Chapter (IAPR-IC) on "Artificial Intelligence and Pattern Recognition Techniques for Computer Vision", April 6-7, pp. 122-127.

Wenyin, L., Dori D. 1999. From Raster to Vectors: Extracting Visual Information from Line Drawings. Pattern Analysis and Applications, 2(2), pp. 10-21.

Wenyin L., Dori D., 1999. Principles Of Constructing A Performance Evaluation Protocol For Graphics Recognition Algorithms. Performance Characterization and Evaluation of Computer Vision Algorithms, Kluwer, pp.97-106.