An Efficient Method to Digitize Polygon Objects of a Black and White Raster Map

K. Kundu[1], **P. Halder**[2], **J. K. Mandal**[3]

[1] Department of Computer Science and Engineering
Govt. College of Engineering & Textile Technology, Serampore, Hooghly, India

[2] Department of Computer Science and Engineering
Purulia Government Engineering College, Purulia, West Bengal, India

[3]Department of Computer Science and Engineering
University of Kalyani, Kalyani, Nadia, West Bengal, India

Abstract

In this article an efficient technique has been presented to digitize polygon objects of a black and white raster map/image. Any polygon segment may be represented by a closed sequence of line segments. Every line segment of a polygon constitutes of black points with RGB value. In this proposed method, initially mouse click operation has to perform on or closer to each of the line segments. If the clicked point is not on the line then eight connected points of each of the click point is calculated and this process may be repeated to obtain the black point. After obtaining the black points for every line segments, each black point are considered as selected points for each of the line segments. Next calculation of successive black points on both sides of the selected point of each of the line segments are performs until it reached to the junction point (i.e. black point has more than three four-connected pixels). Black points on both side of the selected point for a line are merged to obtain the line segment. Finally each of the line segments are merged to obtained the closed polygon. The points of closed segment are minimized using the computation of minimal set of boundary points of segments. Currently many of the existing methods required lot of memory space for the processing to producing the boundary points. This proposed technique efficiently minimizes the complexity to construct the polygon objects.

Keywords: Vector map, Raster map, GIS, Digitization, Spatial data.

1. Introduction

Geographic Information Systems (GIS) first appeared in the 1970s, and were an emerging field that progressively developed on the basis of database, remote sensing, and computer aided drawing (CAD) technologies [2, 5, 6]. From a historical perspective, GIS emerged from maps, thus theories and methods of cartography play an important role in the development of GIS. GIS can be considered a new medium for map information, and has the functions of space information storage, analysis, display and transferring. At the same time, maps are still an important source of data for GIS. GIS tools can process paper maps and convert them into electronic maps displayed on a monitor, so as to satisfy new demands on maps. Therefore, the digitization of maps is an extremely important task, especially for old maps with historic value [3, 4]. Digitization of such maps not only preserve maps in digital form, but also allow maps to become source data for compiling new maps using GIS software. Digitizing is the process of converting geographic features on a raster map into vector format [8, 9, 10]. Raster image represented by collection of pixels, and each spatial position or resolution element has a pixel connected where the pixel value indicates the attribute, such as color, elevation, or an ID number. Vector data are represented by the points, lines and polygon that are geometrically and mathematically related. Point features are stores by the coordinates. Line features are stores as a sequence of coordinate points, where each pair represents a straight line segment.

Various existing methods are available to digitize a polygon objects while majority of them use mouse click, mouse movement, dragging it along the borders of the various parts of the maps or drawn by closed loop inside the segment. Mandal et al. [11] presents polygon objects digitization using mouse drag on the border of the segment but drawback is that higher time is needed to digitize a polygon object and also lot of memory space is required for the processing. The polygon object is also created through mouse click on the border of the segment which is design by Halder et al. [1] but problem is that accuracy is very low and larger time is required to digitize a polygon object. Halder et al. [7] offered a technique for polygon object using only single click inside the segment although weakness is that large memory space is required for generating the boundary points and boundary points contains lot of points which may be minimized. Kundu et al. [15] presents polygon objects vectorization by drawing closed loop inside the segment, but problem is that lot of memory space is required to producing the boundary points and it is a time consuming method due to drawing closed loop inside the segment. To overcome these problems an efficient technique has been proposed in this article. The advantages of the proposed technique are smaller memory space, minimum boundary points and lesser time for generating the polygon objects.

Section 2 of this article describes the proposed scheme. Results are shown in section 3. Comparisons and analysis are presents in section 4. Conclusions are discussed in section 5.

2. Proposed scheme

In this proposed scheme initially black and white raster map is taken. This map composed of the set of polygon segments. Each polygon segment is represented by a closed sequence of line segments. Each line segments constitutes the black points. If noise are exists neighboring of boundary line then it has been remove manually through windows paint or any other software. But if noise exists at the other position then there is no need to remove it. After removing the noise, the proposed technique can be applied to obtain the polygon objects of the map. After that the polygon constructing points are minimized through the computation of minimal set of boundary points of segments which is design by Kundu et al. [15]. Finally database table is created for each of the polygon segments. Figure1 shows the block diagram of the proposed technique. The proposed technique consists of the following four sub procedure.

1. Image correction
2. Spatial data generation
3. Database formation
4. Output generation

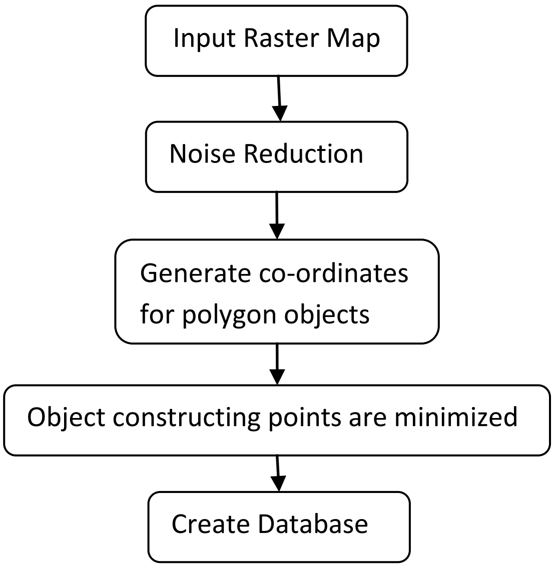
Input Raster Map
Noise Reduction
Generate co-ordinates for polygon objects
Object constructing points are minimized
Create Database

Figure 1: Block diagram of proposed technique

2.1 Image correction

Input raster map is made up of set of polygon segment and each polygon segment composed of line segments. Before running this tool it is required to remove noise from the surrounding of the line segments. But the map has to be checked to verify whether there exists any discontinuity on the boundary of the segments. The rectification carried out through manually for the segments those have discontinuities. If each segment width is more than two pixels then it is required to be transformed into line with one or two pixels width. The correction of image may be performed by using any software such as Windows Paint or Adobe Photoshop.

2.2 Spatial data generation

Actually the Raster map contains visual information about the components of the map where the components cannot be selected. The raster map contains information about pixels and each pixel associated with the RGB value. In vector map the components such as polygon, line and points have database containing the coordinates. Spatial data generation is the process to obtaining the coordinates of every objects of the map. Then the coordinates are stored in the database. To generate the spatial data of every polygon objects of the map eight connected pixels are considered. The eight connected points of a pixel P is shown in Figure 2. From the figure, it is seen that the points are p0, p1, p2, p3, p4, p5, p6, p7 are the eight connected points of the pixel P (x, y). The co-ordinates of the following points are (x, y-1) , (x, y+1) , (x+1, y) , (x-1, y) , (x+1, y-1) , (x-1, y-1) , (x-1, y+1), (x+1,y+1) respectively.

P5	P0	P4
P3	**P**	P2
P6	P1	P7

Figure 2: A pixel (P) and its 3×3 neighborhood

Among eight connected points the black points are considered to digitize the map. The polygon object digitization process is described in section 2.2.1.

2.2.1 Polygon object digitization process

A polygon is represented by a closed sequence of line segments. Each line segments constitutes the black points and each black point associated with the RGB value. Figure 3(a) shows that the polygon segment is composed of line segment1, line segment2 and line segment3 which are represented by blue, green and orange colors respectively. In this proposed technique initially mouse is click on or closer to each of the line segments. If the initial click point is not on the line then calculation of the eight connected points of the click point is required and this process is repeated until a black point among the eight points is obtained. After getting the black point the eight connected pixels are considered to generate the black points of the line segments. The initial selected black point is considered as selected point for a line segment. Figure 3 (b) shows the details representation of the line segment1 which is represented by blue color.

Here initially mouse click is performed on or closer to the line segment1. If click point is not black then calculate the eight connected point of the click point and repeat this step until find a black point. After obtaining the black point it is considered as selected point S1. Similarly the selected points for line segment2, line segment3 are calculated. After obtaining the point S1, the black point P_1 is selected from the neighboring black points of S_1 having maximum black neighbor. After that the status of point P_1 is set as 1. Next, select black point Q_1 with status 0, from the neighbors of S_1 having maximum black neighbor. Subsequently the distance between two points P1 and Q1 is calculated. If the distance is 1 then it indicates that the points P_1 and Q_1 exist in the same side of selected point S_1 which is not desirable.

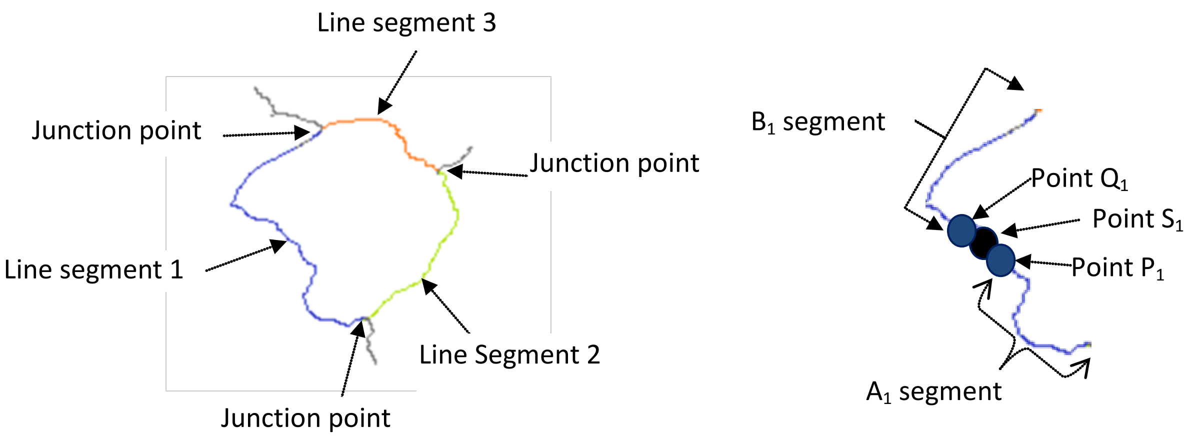

Figure 3: (a) polygon segment (b) Line segment 1 for polygon segment (a)

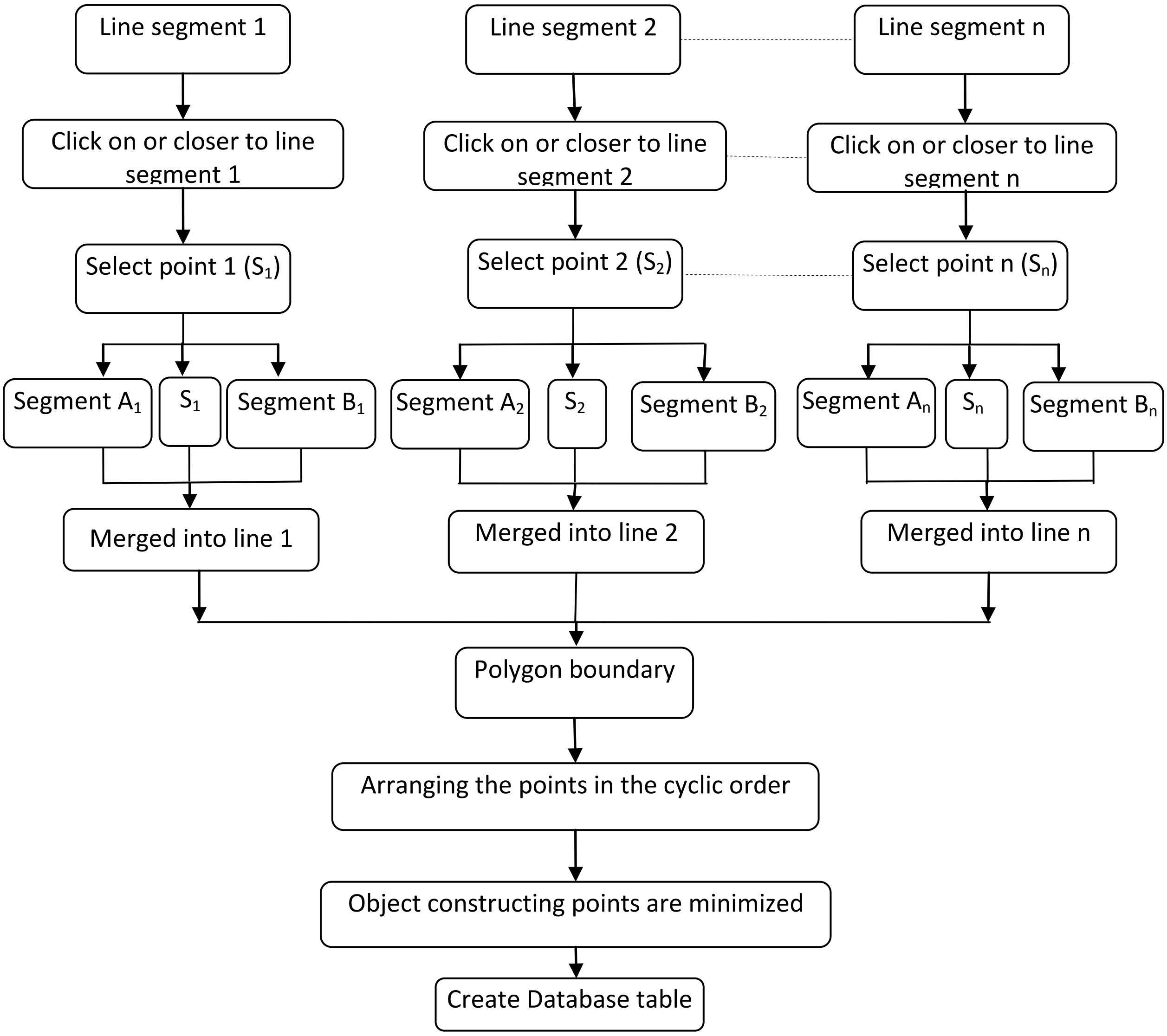

Figure 4: Polygon object digitization process

To obtained the points P_1 and Q_1 in both side of S_1 the status of all the neighbors of P_1 are set as 1 and select black point Q_1 with status 0, from the eight connected points of S_1 having maximum black neighbor. If the distance is not 1 then set the status of all eight neighbors of P_1 as 0. Set status of S_1 as 1, status of eight neighbors of S_1 as 1 and the status of Q_1 as 1. Point S_1 and P_1 are stored in the array A1. Point Q_1 is stored in the array B1. To obtain the points of segment A1, the eight-connected points of the last inserted point in A1 array are calculated. Among the eight points the black point with status 0 having maximum black neighbor is considered as new black point and stored it in array A1 by assigning the status as 1. After that the eight connected points of new black point are calculated. If the new black point has 0 black neighbor then terminate. If the new black point has 1 black neighbor then selects the neighbor and insert it to the array A1. If the new black point has more than 1 black neighbor then find the four connected points of the new black point and if new black point has more than or equal to 3 black neighbor then terminate for junction point. Otherwise calculates the distance between the points having maximum black neighbors and the point having next maximum black neighbors. If the distance is less than 1.1 then insert the point having maximum black neighbor into the array A1 and set the status of the last inserted point as 1. Like this way all the points for segment A1 is calculated. Similarly all the points for B1 segment are calculated and stored in B1 array starting from point Q_1 to the junction point. A1 array and B1 array are merged and obtain the line1 array. Similarly for line2 array and line3 array are obtained. Finally line1 array, line2 array and line3 array are merged and an array named as poly is obtained. The points of Poly array are arranged in cyclic order and stored in poly2 array. After that the boundary points of the polygon segment are minimized and stored in poly3 array using the computation of minimal set of boundary points of segments which is proposed by Kundu et al. [15]. Figure 4 shows the polygon object digitization process for a closed segment.

2.2.1.a Algorithm for polygon object digitization

Input: Closed segment has taken.

Output: Generate polygon object.

Procedure: Initially click on or closer to the line segment 1. If the click point is not black then calculate the eight connected points of click point and repeat this step until any black point is found. After finding the black point it is considered as select point, next calculate two black points such that P_1 and Q_1 which are exists on the opposite sides of the select point. Now, all the black points from point P_1 to the junction point in one direction just opposite to the select point are calculated to obtain the line segment A_1. Similarly from point Q_1 to the junction point in other direction are calculated to obtain the line segment B_1. Points of A_1 and B_1 Segment are merged and obtain the line segment 1. Similar way the line segment 2, line segment 3………line segment n are obtained. Finally merge the points of all the polygon constructing segments (line segment 1, line segment 2, Line segment 3………line segment n) to obtain the closed polygon. Obtaining points are then arranged in proper order and the points are minimized using the computation of minimal set of boundary points of segments which is design by Kundu et al. [15]. The computation of polygon object digitization is given below.

*Step1. Click on or closer to each of line segments. (line segment 1, line segment 2, line segment 3……….
line segment n) of the closed segment.*

Step2. If black points are not found then calculate the eight connected points of each click points and repeat this step until find the black points. Obtaining black points are then considered as *select points (S_1, S_2, S_3 ……S_n) for n number of line segments of the polygon.*

Step3. Select two black points P_1 and Q_1 having maximum black neighbor and existing on both sides of the point S_1. Similarly points P_i and Q_i are obtained for all line segment i(i=2…n).

Step4. All the black points from P_1 to the junction point in one direction just opposite to the S_1 are calculated to obtain the A_1 segment. All black points from Q_1 to the junction point in other direction are calculated to obtain the segment B_1. Similarly all the segments A_i and B_i (i= 2…n) are obtained.

Step5. Points of A_i segment, S_i and points of B_i segment are merged and obtain the points of line segment i (i= 1...n).

Step6. Line segment1, line segment 2 ….line segment n are merged and stores as poly1 array.

Step7. Points of poly1 array are arranged in a cyclic order and stores in another array say poly2 array.

Step8. Points of poly2 array are minimized through the computation of minimal set of boundary points of segments [15] and stores in poly3 array.

Step9. Finally the points of poly3 array produce the polygon object.

2.3 Database formation

In GIS, database is essential for all objects of a vector map. Since this tool produce only polygon objects of a raster map, database management system is necessary to store the obtained spatial data. The spatial data i.e. the coordinates of the boundary points of a polygon segment are stored to the database. Database table is created for each segments and each table contains the coordinates of the boundary points of the segment. Another database table is created to contain the names of each polygonal segment. Actually Oracle 10g is used for the databases.

2.4 Output generation

Output of the proposed technique produce the vector map containing the generated polygon objects. After fetching the co-ordinates from the database table the polygon objects are constructed and displayed in the output page. The output vector map only contains the polygon object because it is the closed segment boundary calculation tool.

3. Results

This tool is developing by Java swing language and Oracle 10g is used for databases. The DBMS is required to store the boundary points of the polygon objects. The input raster map is shown in Figure 6 which composed of set of polygon segments. Each of the polygon segments are represented by closed sequence of line segments. Each closed segment contains line segments are clicked and changing the status of indicated by red color which is shown in Figure 7. The Figure 8 shows the output of the selected parts of the input raster map which is denoted by green color.

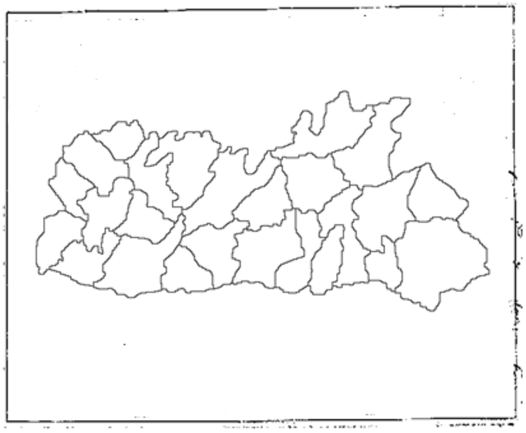

Figure 6: Input raster map

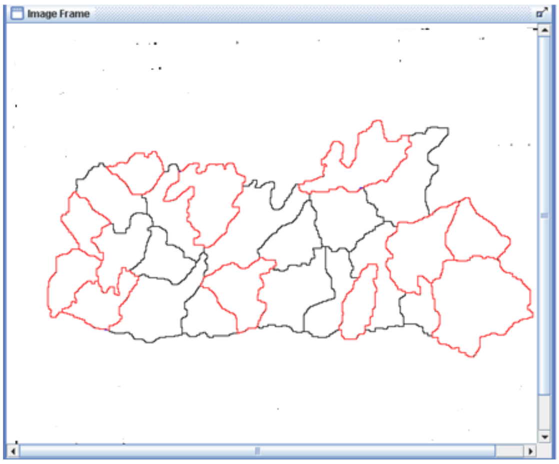

Figure 7: Selected parts of input raster map

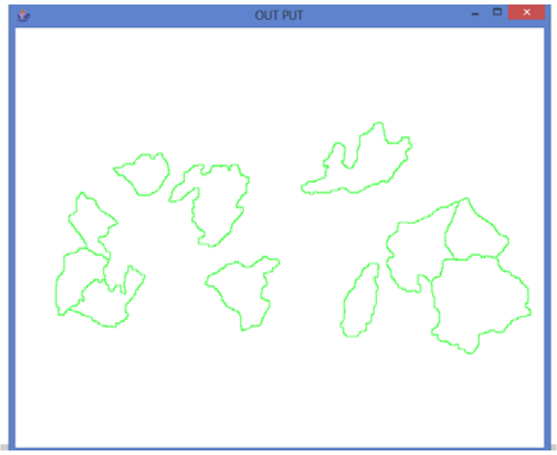

Figure 8: Output of the digitization

4. Comparison and Analysis

The comparisons of the proposed technique with the other existing techniques are given in shown in Table 1. The first technique in table1, proposed by Mandal et al. [11] is a polygon object digitization based on mouse drag. In this method accuracy is low because accuracy depends upon the user and since all dragged points are generated at the time of mouse drag the database size is also very high. The second method is offered by Halder et al. [7] is a digitization technique by automatic boundary points calculation tool but the database size is very high since all the boundary points are stored. Again in this method all the inner points are calculated and compared with the boundary pixels. So, space complexity is also high. The third method is design by Halder et al. [1] is a complete digitization tool for line, polygon and point object using mouse click. Here in this case the accuracy is depends upon the user and it is time consuming in compared with the proposed method. The fourth method is proposed by Kundu et al. [15] is a GIS based tool where digitization process is done by drawing a closed loop inside the segment. In this method accuracy is very high but bulky memory space is required and higher time is needed to producing the boundary points in compared with the proposed technique. In this proposed method polygon objects are digitized where the accuracy is very high as compared with the second method [7]. Here initially mouse click operation is performed on or closer to each of the line segments of the closed segment. Total number of comparisons are required entirely depends on the number of line segments in the closed segment. Closed segment contains boundary points are minimized through the computation of minimal set of boundary points of segments [15] and the points are reduces approximately 50% to 54%. Therefore, database size is decreases.

In the technique polygon objects are digitized through mouse click on or closer to each of the line segments of the closed polygon. Figure 9 shows one output polygon segment of input raster map which composed of the closed sequence of four line segments and junction points. Click points are denoted by C1, C2, C3, and C4. After clicking on each of the line segments, polygon object is produce. In this proposed method computation time entirely depends on the total number of line segments in the closed segment. In this segment only four mouse click operation are required to generate the polygon object. So, computational time decreases in compared with the existing techniques for this segment. After obtaining the boundary points, the same points are minimized through the computation of minimal set of boundary points of segments [15].

Table1. Comparison of the proposed method with existing methods

Sl. No.	Technique	Accuracy	Type of Digitization	Estimated time	Database size	Type of objects
1	Mandal et al. [11]	Low, depends upon the user	Mouse Drag on the border of the segment	Higher than the proposed method	Very High	Polygon
2	Halder et al. [7]	Almost 100%	Automatic, single click in the segment	Very low but internal computation is high	Very High	Polygon
3	Halder et al. [1]	Low, depend upon the user	Mouse Click on the border of the segment	Higher than the proposed technique	Very low, only selected points are stored	Polygon, Line, and Point
4	Kundu et al. [15]	Almost 100%	Closed loop is drawing inside the segment	Smaller than the 1st and 3rd methods. Larger than the 2nd and proposed methods.	Boundary points are minimized to approximately 50% - 54% in comparison with the first and second method.	Polygon and Line
5	Proposed method	Almost 100%	Click on each of line segments in the closed segment	Lower than the 1st, 3rd and 4th methods. Slightly higher than the 2nd method but internal computational cost is smaller than the 2nd method	Same as 4th method	Polygon

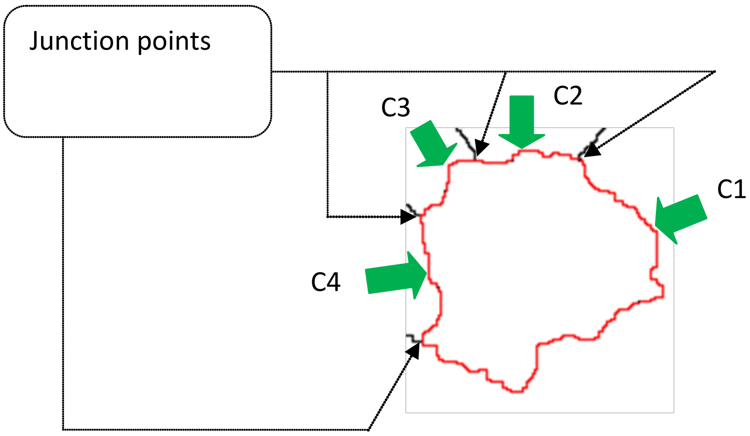

Figure 9: One output polygon segment of input raster map in Figure 6

5. Conclusion

This technique is suitable for the black and white raster maps. If the raster map contains segments with non contiguous boundary then it has to make proper manual rectification with any software such as windows Paint, Adobe Photoshop etc. Image rectification, to get continuous boundary may be time consuming for manual correction. Whenever the line width is more than two pixels, the manual modification is required to transform the width less than three pixels. In this proposed technique one closed raster segment is vectorise by the selection of object constructing line segments using mouse click operation. Therefore, computational time is reduces (as comparison with the other techniques) for generating the boundary points. Boundary points are minimizes through the computation of minimal set of boundary points of segments. Hence, database size is decreases. This proposed method is an efficient technique for digitization of polygon objects with less database size, minimum computational cost and high accuracy.

References

[1] P. Halder, J. K. Mandal, "A GIS anchored Information Retrieval Scheme (GISIRS) based on vectorization of Raster map". International Journal of Engineering Research and Industrial Applications (IJERIA), ISSN: 0974-1518, Vol.-2, No.-III, 2009, pp. 265-280.

[2] AP Dal Poz, RB Zanin, GM d Vale, "Automated extraction of road network from medium-and high-resolution images". Pattern Recog. Image Anal. 16(2), 2006, pp. 239–248.

[3] P. Halder, J.K. Mandal, "A Polygon Based Thematic Map Representation Technique Using Database Management System". A National Conference on Emerging Trends in Information Technology, P.E.S. College of Engineering, Aurangabad, 28-29 January 2008, pp. 1-6.

[4] J.K. Mandal, S. Das, A. Nath, and P. Sarkar, "A Pest Management System of Tea Garden Using GIS Based Database Management System", Proceedings of First International Conference on Information Technology, INTL-INFOTECH 2007(ISSN-0973-6824), Haldia: Haldia Institute of Technology, 2007, pp. 261-264.

[5] J.B. Mena. "State of the Art on Automatic Road Extraction for GIS Update: a Novel Classification". Pattern Recognition Letters, 24(16):2003, pp. 3037-3058.

[6] Straub, B., Wiedemann, C., and Heipke, C., "Towards the automatic interpretation of images for gis update,"International Archives of Photogrammetry and Remote Sensing 33(B2; PART 2), 2000, pp. 525–532.

[7] P. Halder, J. K. Mandal, S. Mal, "An Efficient Technique to Digitize Polygon Segments of a Map/ Image (ETDPS)". IJITKM, ISSN: 0973-4414, Volume 2, No. 2, 2009, pp. 313-316.

[8] X. Wu, R. Carceroni, H. Fang, S. Zelinka, and A. Kirmse." Automatic alignment of large-scale aerial rasters to road-maps". In Proceedings of the 15th ACM GIS, 2007, pp. 1–8.

[9] Y.-Y. Chiang and C. A. Knoblock. "Automatic extraction of road intersection position, connectivity, and orientations from raster maps". In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2008, pp. 1–10.

[10] P. Halder, J. K. Mandal, S. Mal, "A Decision Support System Based on 2D & 3D Graphical Representation on Segments of Remote Sensing Image/ Local Map". IEEE

International Advance Computing Conference (IACC 2009), ISBN: 978-981-08-2465-5, Thapar University, Patiala, India, 2009, pp. 2172-2176.

[11] J.K. Mandal, S. Das, A. Moitra, “A GIS Based Database Oriented Vectorization of Tea Garden Image for Decision Making”, ACVIT- 2007, Aurangabad, 2007, pp. 839-846.

[12] Collab.teldap.tw/digitalLibrary/Maps GIS.pdf online access date: 12.10.2015.

[13] Ecksten,R. Loy, M. and Wood, D. (2001),Java Swing, Sebastpol: O’Reilly & Associates, Inc.

[14] Schildt,H. (2005), The Complete Reference-Java, New Delhi: Tata McGrow-Hill Publishing Company Limited.

[15] K. Kundu, P. Halder, J. K. Mandal, "A GIS Based Tool to Digitize Polygon and Line Segments of a Black and White Map". Journal of Remote Sensing & GIS (JoRSG) ISSN: 2230-7990, Vol. - 7, No. - 1, 2016, pp. 27-41.