# Digitization of Polygon Objects in a Raster Map based on the Direction

K. Kundu
Department of Computer Science and Engineering
Govt. College of Engineering & Textile Technology,
Serampore, Hooghly, India

P. Halder
Department of Computer Science and Engineering
Purulia Government Engineering College,
Purulia, West Bengal, India

J. K. Mandal
Department of Computer Science and Engineering
University of Kalyani, Kalyani,
Nadia, West Bengal, India

## ABSTRACT

This article represents the digitization of polygon objects from a black and white raster map. Based on the selection of direction mouse click operation is perform on the boundary of the polygon object or closer to the boundary. If the initial click point is not black then eight connected points of the click point are compared to obtain the nearest black boundary point and it is considered as starting point (i.e. sp point). After obtained the starting point (sp), eight connected points of the starting point (SP) are calculated and the appropriate black point from the eight connected points is selected according to the rules of the direction and this process is repeated until it reached to the starting point (SP). This proposed technique requires less computation with minimal memory requirement and boundary points are obtained in a single click. The boundary points are minimized approximately 52% through the computation of minimal set of boundary points of a segments. Comparisons with existing reveals that the proposed technique is outperform than existing.

## Keywords

GIS, raster map, vector map, digitization, clockwise and anticlockwise direction.

## 1. INTRODUCTION

A geographic information system (GIS) is a computer-based tool for mapping and analyzing geographic features and it exists on the Earth surface. In general, a GIS [1-3] provides facilities for data capture, data management, data manipulation and analysis. Map is the recognition and representation of the location of geographic features on the landscapes of the Earth surface. A map is composed of different geographic features such as points, lines, and polygon. Each feature is defined both by its location in space (with reference to a coordinate system), and by its characteristics (generally referred to as attributes). Digitization [4-6] is the process of converting geographic features on a raster map into vector format. Traditionally spatial data have been stores and presents in the form of a map. Spatial data models are used for storing the spatial location of geographic features in a database. Raster data model consists of the grid cell where the geographic area is divided into cells marked by row and column. Vector storage refers to the use of vectors (directional lines) to represent a geographic feature. It is characterized by the use of sequential points or vertices to define a linear segment. Each vertex consists of an X coordinate and a Y coordinate.

To digitize a polygon object most of the existing methods are use mouse movement, mouse clicked operation inside the segment and on the border of the segment or mouse dragging along the border in different sections of the map or drawing a closed loop inside the segments. Halder et al. [7] presents a polygon object digitization using mouse click operation on the border of the segment but difficulty is that exactness is very poor due to manual mouse click operation on the border of the segment and it is very time consuming process. The polygon objects may be digitized through mouse drag on the border of the segment which was designed by Mandal et al. [8]. The drawback of this method is that higher time is required to digitize a polygon object and also it is very difficult to produce the accurate digitization. Halder et al. [9] offers to digitize of polygon object using single mouse click operation inside the segment. But, in this method higher memory space is required for creating the boundary points. The polygon boundary contains all the existing points which may not be needed to store in the database and it is required to be minimized. Another method introduced by Kundu et al. [12] presents the polygon object digitization by drawing closed loop inside the segment, but drawback is that much memory space is required in compared with the proposed method for generating the boundary points and time consuming process due to the drawing closed loop inside the segments. Kundu et al. [13] proposed a polygon object is digitization technique through mouse clicking on the line segment of a closed segment, but problem is that to producing the boundary points more than one mouse clicked is required, hence it is the time consuming process in compared with the proposed technique. To overcome these problems an efficient technique has been introduced in this paper. The elegant advantages of the proposed method are smaller memory space, minimization of boundary points and smaller time is required for producing the polygon objects.

Section 2 of this paper is describes the proposed technique. Results are discussed in section 3. Comparisons and analysis are presented in section 4. Conclusions are discusses in section 5.

## 2. PROPOSED TECHNIQUE

In this proposed technique black and white raster map is taken. This raster map or image composed of grid cells where the geographic area is divided into cells and marked by row and column. Each cell associated with a pixel value. The conceptual raster map or image is continuous and also represents the discrete geographic features like point, line, and polygon. Each polygon feature is represented as a group of connected cells portraying shape. If any boundary discontinuity exists on any segment of the raster map then manual correction is needed to remove the discontinuity by using any software such as Windows Paint or Adobe Photoshop software. For image rectification if any noise exists inside the segment then it is not required to remove the noise. Spatial data are generated through the four points (i.e. East, North, South, and West) with respect to the center pixel. After removal of noise, proposed method is applied to generate the boundary points of each of the segment. The obtained boundary points are minimized using the computation of minimal set of boundary points of segments [12]. Each digitized polygon segment of the map is store on a separate database table. Fig 1 illustrates the block diagram of the proposed technique which consists of the following steps:

1. Image Rectification
2. Computation of Spatial Data
3. Database Creation
4. Output Generation

## 2.1 Image Rectification

Rectification of image is required before running this tool. If the input raster map contains any discontinuity on the boundary of the segments then the discontinuity is to be removed. The boundary line rectification was needed to convert the width of the line less than three pixels in the technique of Kundu et al. [12]. But in this technique such rectification is not needed except for the initial clicked point. The initial mouse click point must be on the boundary or closer to the boundary line with width of one pixel.

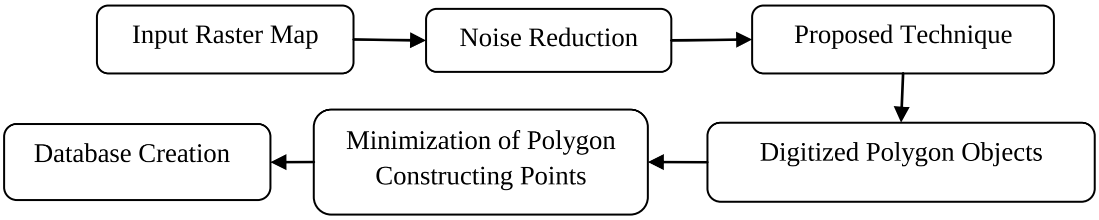
Input Raster Map
Noise Reduction
Proposed Technique
Digitized Polygon Objects
Minimization of Polygon Constructing Points
Database Creation

**Fig 1: Block diagram of proposed method**

## 2.2 Computation of Spatial Data

To generate the spatial data for every polygon objects of the raster map eight connected points in four directions are considered. These directions are shown in Fig 2 and which are East, West, North, and South. The eight connected points of a pixel P is shown in Fig 3. From the figure, it is observed that the points are E, W, N, S, NE, NW, SW, SE are the eight connected points of the pixel P (x, y). The co-ordinates of the following points are (x+1, y) , (x-1, y) , (x, y-1) , (x, y+1) , (x+1, y-1) , (x-1, y-1) , (x-1, y+1), (x+1,y+1) respectively.

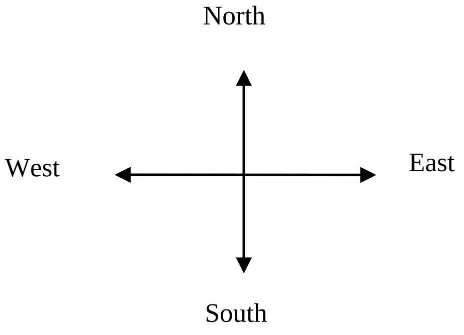

**Fig 2: Four directions**

| | | |
|---|---|---|
| NW(x-1,y-1) | N(x,y-1) | NE(x+1,y-1) |
| W(x-1,y) | P(x,y) | E(x+1,y) |
| SW(x-1,y+1) | S(x,y+1) | SE(x+1,y+1) |

**Fig 3: Eight connected points of P(x, y) pixel**

### *2.2.1 Polygon Object Digitization Process*

In this proposed method initially the direction (i.e. clockwise or anticlockwise) are selected to trace the boundary line of the polygonal segment at the time of the digitization process. Mouse click operation is performs on the boundary of digitizing polygon object. If the initial click point is not black (not on the line) then eight connected point of the click point are calculated and this process will be repeated until a black point (i.e. sp) on the boundary is obtained. The Fig 4 shows the details representation of the polygon segment. The mouse click operation is performed on the polygon segment which is denoted by red color arrow and junction points are shown in blue color arrow. The clockwise direction is represented by black color arrow and the reverse side of the clockwise direction is called the anticlockwise direction.

For clockwise direction, subsequent points selection operation is performs using following rules: if the un traverse North point of the sp (at the beginning) point is black then select the point and store it otherwise check the East point of the sp point and if it is black then select the point and store it otherwise select the West point and store it. When junction point is reached next selection operation is performs using following rules: if preceding point of the junction point is in North, East of the junction point is checked and if it is black then select the point and store it otherwise check the South point and if it is black then select the point and store it otherwise select the West point and store it. If preceding point of the junction point is in South then checked the West point of the junction point and if it is black then select the point and store it otherwise checked the South point and if it is black then select the point and store it otherwise select the East point and store it. If preceding point of the junction point is in West then checked the North point of the junction point and if it is black then select the point and store it otherwise checked the West point and if it is black then select the point and store it otherwise select the South point and store it. If preceding point of the junction point is in East then checked the South point of the junction point and if it is black then select the point and store it otherwise checked the East point and if it is black then select the point and store it otherwise select the North point and store it.

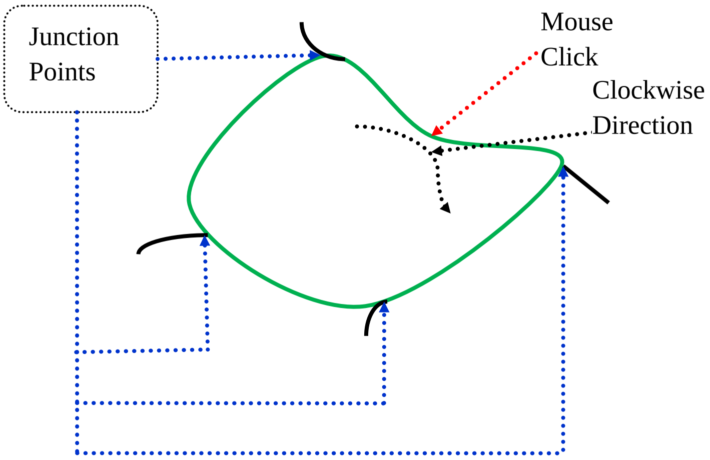

**Fig 4: Polygon Segment**

For anticlockwise direction, subsequent points selection operation is performs using following rules: if the un traverse South point of the sp (at the beginning) point is black then select the point and store it otherwise checked the East point and if it is black then select the point and store it otherwise select the West point and store it. When junction points is reached next selection operation is performs using following ways. If preceding point of the junction point is in South then checked the East point of the junction point and if it is black then select the point and store it otherwise checked the South point and if it is black then select the point and store it otherwise select the West point and store it If preceding point of the junction point is in East then checked the North point and if it is black then select the point and store it otherwise check the East point and if it is black then select the point and store it otherwise select the South point and store it. If preceding point of the junction point is in West then checked the South point of the junction point and if it is black then select the point and store it otherwise checked the West point and if it is black then select the black point and store it otherwise select the North point and store it If preceding point of the junction point is in North then checked the West point of the junction point and if it is black then select the point and store it otherwise checked the North point and if it is black then select the point and store it otherwise select the East point and store it.

Algorithm for digitization of a polygon object in clockwise direction is given in section 2.2.1.a.

### *2.2.1.a. Algorithm: Clockwise_Direction (K, Clock, SP, T)*

K =total no. of black co-ordinate points.

Clock is an array which contains the black co-ordinate points and size is 0..K-1.

SP is the initial black point.

T is the neighboring black points.

East, West, North and South are the four main directions.

K=0

Input: Raster map, click on the closed segment or near to the closed segment.

Output: Vector-polygon object.

Method: Gradually trace from initial select black point to the successive next black point in the clockwise direction until it reached to the initial black point.

```
Step1. If the initial click point is not black then calculate the eight connected points of the click point and repeat this step until the black point is obtained and select the point as SP.
        Clock[K]=SP.
        Set the status of SP point as 1 (assumed first point).
Step2. If the North point of SP is black then K++,
        Clock[K]=North point and set the status of
        point as 1 else
        If the East point of SP is black then K++,
        Clock[K]=East point and set the status of point
        as 1 else
        Clock[K]=West point and set the status of
        point as 1.
Step3. Set the status of SP point as 0.
Step4. Repeat steps 5 to10 while SP≠ Clock[K]
        otherwise exit.
Step5. Set T= black neighbors of Clock[K].
Step6. Repeat step6   if T<3 otherwise go to step 7.
        If the un traverse North point of Clock[K] is black then K++, Clock[K]=North point  and set the status of the point as 1 and go to step 4 else if the un traverse East point of Clock[K] is black then K++ ,Clock[K]=East point and set the status of point as 1 and go to step 4 else K++, Clock[K]=West point and set the status of point as 1 and go to step 4.
Step7. If Clock[K-1]=North then
    a.  If the East point of Clock[K] is black then K++, Clock[K]=East point and set the status of point as 1 and go to step 4 else
    b.  If the South point of Clock[K] is black then K++, Clock[K]=south point and set the status of point as 1 and go to step 4 else
```

c. K++, Clock[K]=West point and set the status of point as 1 and go to step 4.

Step8. If Clock[K-1]=South then

a. If the West point of Clock[K] is black then K++, Clock[K]=West point and set the status of point as 1 and go to step 4 else
b. If the South point of Clock[K] is black then K++, Clock[K]=South point and set the status of point as 1 and go to step 4 else
c. K++, Clock[K]=East point and set the status of point as 1 and go to step 4.

Step9. If Clock[K-1]=West then

a. If the North point of Clock[K] is black then K++, Clock[K]=North point and set the status of point as 1 and go to step 4 else
b. If the West point of Clock[K] is black then K++, Clock[K]=West point and set the status of point as 1 and go to step 4 else
c. K++, Clock[K]=South point and set the status of point as 1 and go to step 4.

Step10. If Clock[K-1]=East then

a. If the South point of Clock[K] is black then K++, Clock[K]=South point and set the status of point as 1 and go to step 4 else
b. If the East point of Clock[K] is black then K++, Clock[K]=East point and set the status of point as 1 and go to step 4 else
c. K++, Clock[K]=North point and set the status of point as 1 and go to step 4.

Algorithm for digitization of a polygon object in anticlockwise direction is given in section 2.2.1.b.

### *2.2.1.b. Algorithm: AntiClockwise_Direction (J, Anticlock, SP, T)*

J = total no. of black co-ordinate points.

Anticlock is an array which contains the black co-ordinate point and size is 0..J-1.

SP is the initial black point.

T is the neighboring black points.

East, West, North and South are the four main directions.

J=0

Input: Raster map, click on the closed segment or near to the closed segment.

Output: Vector-polygon object.

Method: Gradually trace from initial select black point to the successive next black point in the anticlockwise

direction until it reached to the initial black

point.

Step1. If the initial click point is not black then calculate the eight connected points of the click point and repeat this step until obtain the black point and select the point as SP.

Anticlock[J]=SP.

Set the status of SP point as 1 (assume).

Step2. If the South point of SP is black then J++,

Anticlock[J]=South point and set the status of

point as 1 else if the East point of SP is black

then J++, Anticlock[J]=East point and set the

status of point as 1

else Anticlock[J]=West point and set the status

of point as 1.

Step3. Set the status of SP point as 0.

Step4. Repeat steps 5 to10 while SP≠ Anticlock[J]

otherwise exit.

Step5. Set T= black neighbors of Anticlock[J]

Step6. Repeat step6 if T<3 otherwise go to step 7

If the un traverse South point of Anticlock[J] is black then J++, Anticlock[J]=South point and set the status of point as 1 and go to step 4 else if the un traverse East point of Anticlock[J] is black then J++, Anticlock[J]=East point and set the status of point as 1 and go to step 4 else J++, Anticlock[J]=West point and set the status of point as 1 and go to step 4.

Step7. If Anticlock[J-1]=South then

a. If the East point of Anticlock[J] is black then J++, Anticlock[J]=East point and set the status of point as 1 and go to step 4 else

b. If the South point of Anticlock[J] is black then J++, Anticlock[J]=South point and set the status of point as 1 and go to step 4 else

c. J++, Anticlock[J]=West point and set the status of point as 1 and go to step 4.

Step8. If Anticlock[J-1]=East then

a. If the North point of Anticlock[J] is black then J++, Anticlock[J]=North point and set the status of point as 1 and go to step 4 else

b. If the East point of Anticlock[J] is black then J++, Anticlock[J]=East point and set the status of point as 1 and go to step 4 else

c. J++, Anticlock[J]=South point and set the status of point as 1 and go to step 4.

Step9. If Anticlock[J-1]=West then

a. If the South point of Anticlock[J] is black then J++, Anticlock[J]=North point and set the status of point as 1 and go to step 4 else

b. If the West point of Anticlock[J] is black then J++, Anticlock[J]=West point and set the status of point as 1 and go to step 4 else

c. J++, Anticlock[J]=North point and set the status of point as 1 and go to step 4.

Step10. If Anticlock[J-1]=North then

a. If the West point of Anticlock[J] is black then J++, Anticlock[J]=West point and set the status of point as 1 and go to step 4 else

b. If the North point of Anticlock[J] is black then J++, Anticlock[J]=North point and set the status of point as 1 and go to step 4 else

c. J++, Anticlock[J]=East point and set the status of point as 1 and go to step 4.

## 2.3 Database Creation

In GIS, database is important for all objects of a vector map. In this tool Oracle 10g is used for the databases which store the spatial co-ordinate points. Since this tool produce only polygon objects of a raster map, database management system is essential to stores the obtained spatial data of the polygon objects. The spatial data i.e. the coordinates of the boundary points of a polygon segment are stores to the database. For each segment one database table is created which contains the coordinates of the boundary points of the segment. Another database table is required to stores the name of each polygonal segment.

## 2.4 Output Generation

Output of the proposed method is to create the vector map which contains digitize co-ordinate points of each of the segment. Vector map contains set of digitize polygon segment. On fetching the co-ordinates from the database table the polygon objects are constructed and displayed in the output page.

## 3. RESULTS

This tool is implemented using Java Swing language and Oracle 10g databases are used for stores the information of the segments. Each raster map or image composed of grid cell where the geographic area is divided into cells and marked by row and column. Each cell associated with the pixel value. The conceptual raster map or image is continuous and also represents the discrete geographic features like point, line, and polygon. Each polygon feature is represented as a collection of connected cells portraying shape. The input raster map is shown in Fig 5. The Fig 6 represents the output of the selected parts of the input raster map which is mark by red color. Output of the selected segments of the input raster map is shown in Fig 7 which indicates by the green color.

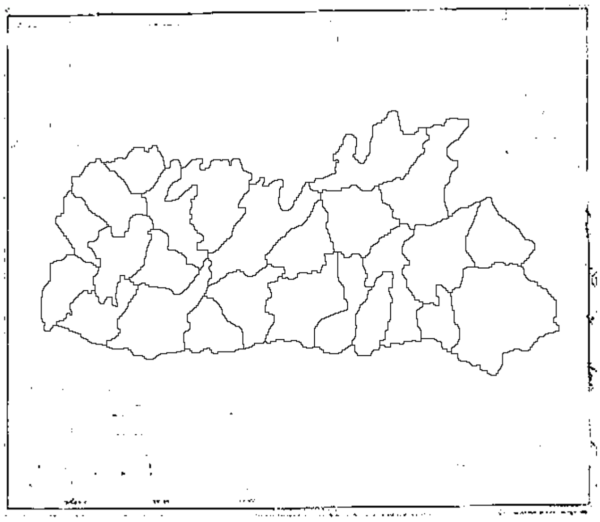

**Fig 5: Input Raster map**

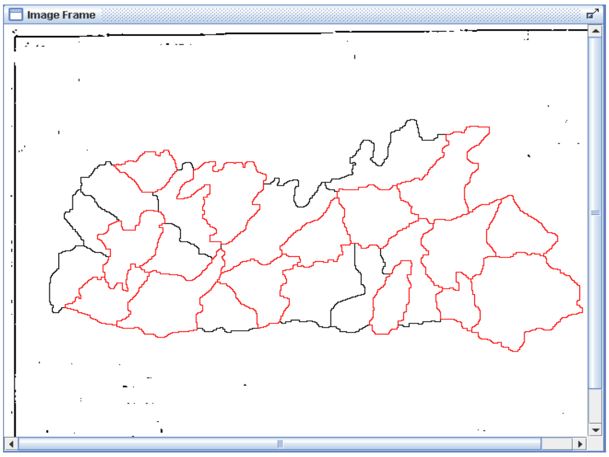

**Fig 6: Selected parts of input raster map**

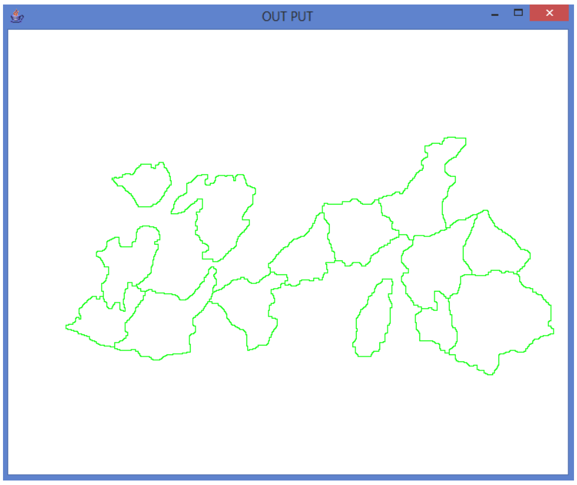

**Fig 7: Output of the selected parts of input raster map**

## 4. COMPARISONS AND ANALYSIS

The comparison of the proposed method with the existing methods is shown in table 1. The method design by Halder et al. [7], presents the polygon object digitization through the mouse click on the border of the segment. It is a time consuming method and accuracy is very low as the mouse click operation is performed on the border of the segments manually. But in the proposed method database size is very low as selected points are stored. Again the method proposed by Mandal et al. [8], represents the polygon object digitization using the mouse drag on the border of the segment. In this method accuracy of the output is low and it is a time consuming technique and much more memory space is required for producing the boundary points. In the method proposed by Halder et al. [9] the polygon object is digitize using the single mouse click inside the segment. Though it is not time consuming method but much more memory space and internal computation is required for generating the boundary points in compared with the other techniques which is mentioned in table 1. The method design by Kundu et al. [12], presents digitization of polygon objects by drawing closed loop inside the segment. In this technique the accuracy is very high and database size is very low but efficiency entirely depends on the drawing a closed loop inside the segment (Drawing closed loop how much closer to the segment). In this proposed technique initially select the direction of tracing pixels along the boundary of a segment either in clockwise or anticlockwise direction. After selecting the direction mouse click operation is performed on the border of the segment and produce the boundary points. The calculated boundary points are minimized by the computation of minimal set of boundary points of segments [12]. So, the proposed method is less time consuming due to single mouse click operation is needed to produce the boundary points and accuracy is very high, low memory space and less internal computation is required.

In this proposed technique initially single mouse click operation is performed on the polygon segment or closer to the polygon segment which is marked by the C1 in Fig 8 where the junction points are also shown. To digitize a polygon object in this technique single mouse click operation is required. The polygon constructing points are obtained by considering only the black points on the boundary of the raster segment. So, computational cost is reduced compared to the other existing techniques for all the segments. Boundary points are minimized through computation of minimal set of boundary points of segments [12]. For this segment (Fig 8) total numbers of boundary points are exists 508 points. But using the computation of minimal set of boundary points of segments [12] subsequent boundary points are produces using only 252 points. Therefore, approximately 50% boundary points are reduces for this segment. The Fig 9 shows the output of the selected parts of input raster map with labeling from S1 to S10. Table 2 display the estimation of total no. of boundary points, total no. of points after minimization and % of reduction. From the experimental results it is observed that approximately 52% points are reduced for the 10 segments (S1 to S10) in Fig 9

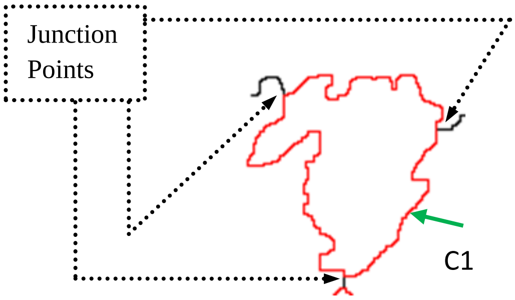

**Fig 8: Output of the selected part of input raster map**

**Table 1: Comparisons of the proposed technique with the other existing method**

| Sl. No. | Method | Exactness | Digitization Type | Estimated time | Database size | Object Type |
|---|---|---|---|---|---|---|
| 1 | Halder et al. [7] | Low, depends upon the user | Mouse Click on the border of the segment. | Higher than the proposed method. | Very low, only selected points are stored. | Polygon, Line, and Point |
| 2 | Mandal et al. [8] | Almost 100% | Mouse Drag on the border of the segment. | Higher than the proposed technique. | Very High | Polygon |
| 3 | Halder et al. [9] | Low, depend upon the user | Automatic, single mouse click in the segment. | Very low but internal computation is high. | Very High | Polygon |
| 4 | Kundu et al. [12] | Almost 100% | Closed loop is drawn inside the segment. | Lower than the 1st , 2nd and 3rd method | Boundary points are minimized to approx. 52% in compared with the 1st and 2nd method. | Polygon and Line |
| 5 | Proposed technique | Almost 100% | Automatic, single mouse click on the segment. | Very fast in compared with the other existing techniques. | Same as 4th method. | Polygon |

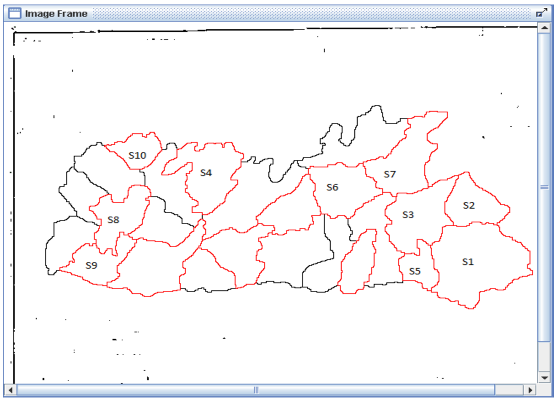

**Fig 9: Output of the selected parts of input raster map with label (S1 to S10)**

**Table 2: Estimation of total no. of points, total no. of points on minimization and % of reduction for the segments S1to S10**

| Sl. No. | Segment | Total No. of Points | After Minimization<br>(No. of Points) | Approx.<br>% of Reduction |
|---|---|---|---|---|
| 1 | S1 | 500 | 216 | 57% |
| 2 | S2 | 298 | 160 | 46% |
| 3 | S3 | 441 | 208 | 53% |
| 4 | S4 | 508 | 252 | 50% |

| | | | | |
|---|---|---|---|---|
| 5 | S5 | 240 | 91 | 62% |
| 6 | S6 | 347 | 162 | 53% |
| 7 | S7 | 483 | 246 | 49% |
| 8 | S8 | 396 | 175 | 56% |
| 9 | S9 | 388 | 196 | 49% |
| 10 | S10 | 241 | 133 | 45% |

## 5. CONCLUSIONS

This tool is suitable for only black and white raster map. If any segment in the raster map contains any discontinuity then continuation has been done by using the image processing or using the Windows Paint or Adobe Photoshop software. If the neighboring of the segments or inside the segments contain any noise then these noise are not required to remove except the initial clicked point. The limitation of this tool is that initial mouse click point must be on the boundary line with width of one pixel. In this proposed technique to produce the boundary points of a segment only single mouse click operation is required. The polygon constructing points are calculated by tracing the boundary points either in clockwise or anticlockwise direction. After digitizing the polygon object, the same points are minimized using the computation of minimal set of boundary points of segments [12]. Therefore, this technique is efficiently minimized the boundary points and reduces the time complexity due to the less number of comparisons required to produce the boundary points.

## 6. ACKNOWLEDGEMENT

The authors express sincere gratitude to all faculty and staff members of the Department of Computer Science and Engineering, University of Kalyani for their supports. The authors also thankful to Head and all faculty members of Department of Computer Science and Engineering and Department of Information Technology of Government College of Engineering and Textile Technology, Serampore for their help to carry out the work.

## 7. REFERENCES

[1] 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.

[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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] K. Kundu, P. Halder, J. K. Mandal,” An Efficient Method to Digitize Polygon Objects of a Black and White Raster Map”. International Journal of Computer Science Issues (IJCSI), ISSN: 1694-0784, Vol. - 13, No. - 4, 2016, pp. 90-99.