

# A NOVEL REGION FILLING ALGORITHM FOR DISCONTINUOUS CONTOURS

Arshad Javeed<sup>1</sup>

<sup>1</sup>Student, Dept. of Electronics and Communication, Sri Jayachamarajendra College of Engineering (SJCE), Karnataka, India

## Abstract

*This is a new algorithm which can be applied to contours having discontinuous boundaries in order to obtain its 1-pixel thickness equivalent and complete the shape of the contour which can be followed by region filling operation. The conventional thinning algorithm often modifies the shape of the contour and the obtained skeleton will not resemble the original image and hence the thinning operation is not used in the proposed algorithm.*

*After reducing the thickness of the boundary to 1 pixel, the shape is completed by traversing along the boundary and joining each black pixel (if there is no black pixel in N8 (p)) to its nearest black pixel. The completion of the shape is necessary because if region filling algorithm is applied directly it will result in blackening of the entire image or may have no effect depending on the algorithm being used. Hence after linking the discontinuities an existing region filling algorithm such as Boundary Fill is employed to fill the region.*

**Keywords:-** region filling; 1-pixel thickness; boundary traversing; shape completion; discontinuous contours; linking pixels; distance transform;

\*\*\*

## 1. INTRODUCTION

Region filling is one of the basic morphological operations that involves filling the region bounded by the contour.

The region filling operation is widely used in medical image processing, object detection and computer aided graphic design and other fields. Presently there are many region filling algorithms having their own advantages. Some of the widely used algorithms are:

- Seed fill
- Parity Check
- Line Coherence

In Seed Fill [1] algorithm a seed point (point interior of the shape) is selected and the filling process starts at this point. The points adjacent to the known points are identified and filled. This process is continued until the boundary is reached. For the algorithm to work correctly, it must simultaneously store the filled frames in memory for comparison later on. This increases the space complexity.

The parity check region filling algorithm is based on the even-odd parity rule. Here, any selected pixel is considered to be inside the contour if the line segment drawn from the pixel across the image intersects the black boundaries odd number of times. Otherwise, it is considered to lie outside the contour. Only those pixels that are found to lie inside the contour are shaded. The obvious problem here is that the presence of stray line segments which are not related to the shape at all would also be detected as edges of the shape and would result in inappropriate filling. There are a number of parity check algorithms to counter this problem as described

by Pavlidis[2]. Scanline coherence algorithm assumes that the neighboring pixels on scanned line tend to lie in the same region and the region is filled by scanning each line. Span based Seed fill is similar to line coherence algorithm except that the span pixels in each row are determined after filling the current row.

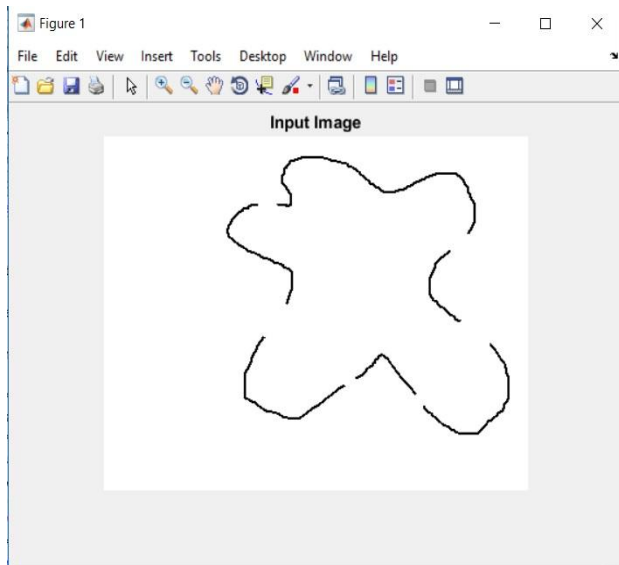
XunGen Li[3] has developed a new region filling algorithm based on the concept of chain codes and has compared the performance of the seed fill algorithms.

The algorithms mentioned above may have their own advantages, but they work well only when the boundary of the shape is continuous. If the discontinuities are small (say 3 or 4px wide), the problem can be solved by simply performing image closing operation before region filling. But in practical cases, this may not be the situation and region filling of such image leads in blackening of the entire image or may have no effect on the input image depending on the algorithm being used.

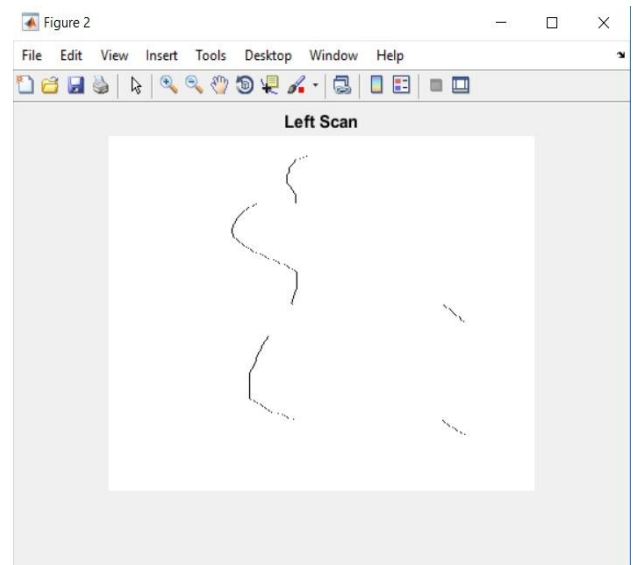
The scope of this paper is to tackle this problem and to demonstrate region filling of disrupted contours using MATLAB.

## 2. ALGORITHM

The working of the algorithm can be better understood by considering the output at each step of the algorithm.



**Fig 1: Original Image**



**Fig 2: Result of left Scan of Input Image**

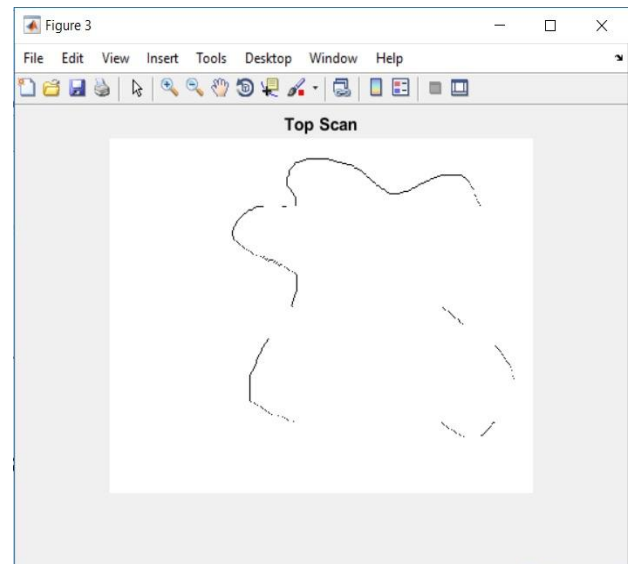
Consider the input image as shown in Fig. 1. It can be observed the discontinuities present along the boundary of the contour cannot be linked by simply applying image closing operation or any other image processing technique. This is because of larger width of the discontinuities.

The first step is to reduce the thickness of the boundary to 1 pixel. As mentioned earlier, applying thinning operation ruins the shape of the contour. Hence a different approach is used here.

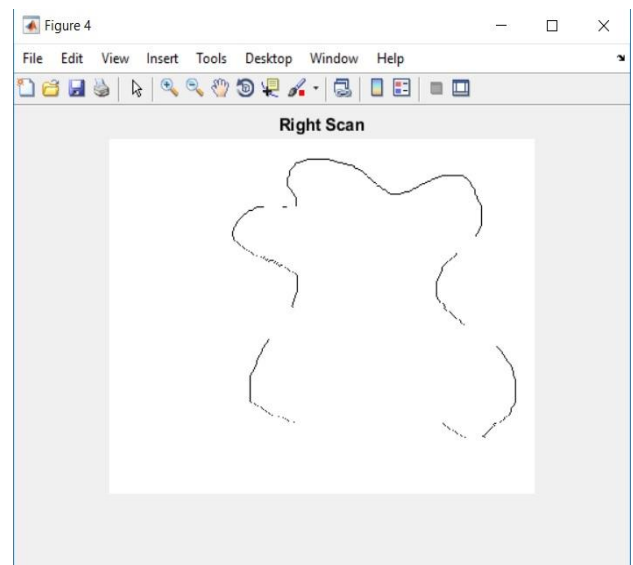
## 2.1 Reducing the Thickness of the Boundary to 1px

The reason for reducing the thickness of the boundary to 1 pixel is to ease the process of traversing the boundary. 1-pixel thickness implies that each black pixel 'p' will have at most 1 black pixel in its  $N_8(p)$  ( $N_8(p)$  is the set representing the 8 neighbors of the pixel p). This makes the task of traversal much easier.

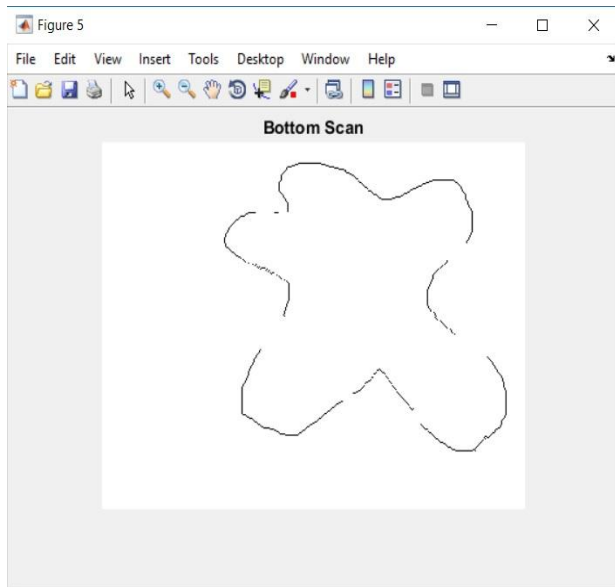
First, each row of the image is scanned from left to right. On encountering a black pixel, the remaining pixels in the row are skipped. The result of this operation is shown in Fig. 2. Next, the process is repeated by scanning each column of the input image from top to bottom. The result of this operation is shown in shown in Fig. 3. Similarly, the image is scanned from right to left (row-wise) and bottom to top (column wise). The results are shown in Fig. 4 and Fig. 5. Scanning from different directions helps to locate the pixels that may not be visible from one direction. This helps in tracing the boundary more accurately.



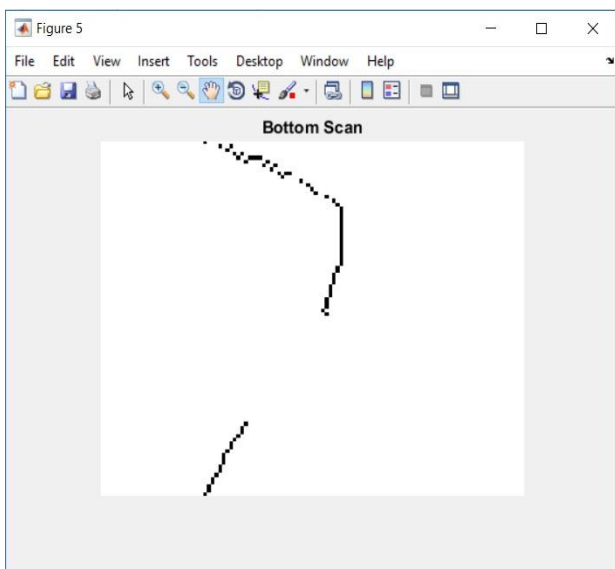
**Fig 3: Result of top Scan of Input Image**



**Fig 4: Result of right Scan of Input Image**



**Fig 5:** Result of bottom Scan of Input Image



**Fig 6:** Zoomed-in version of the final scan

From the Figure 6 it is clear that the thickness of the boundary has been reduced to 1 pixel. However, there are some discontinuities that are still present in the image. These discontinuities are linked in the next step of the algorithm. Also while scanning from different directions, the MATLAB code was written such that any two scans do not produce pixels that overlap each other.

## 2.2 Linking the Discontinuities

Initially, 2 copies of the Image shown in Figure 5 are made. Let 'A' denote the matrix representation of the first copy of the image and 'B' the matrix representation of the second copy of the image.

It is important to note that

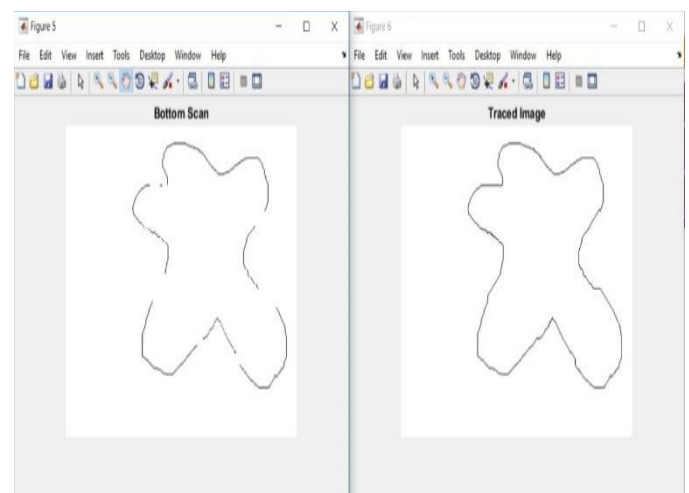
$$A = \sim B \text{ i.e. } A \text{ is complement of } B.$$

Along the matrices A and B, an  $M \times N$  matrix is created and is constantly updated to keep track of the visited black pixels. This matrix helps in visiting the new pixels every time a pixel along the boundary is addressed.

The operations performed in this step are as follows:

1. A random white pixel is selected by scanning the matrix 'A' from left to right. This corresponds to selecting a black pixel from matrix B. Let 'p' denote the detected pixel. The pixel p is marked as visited.
2. Check for white pixels in  $N_8(p)$  in matrix A which corresponds to finding black pixels in B.
3. If a black pixel is found which has not been visited at pixel (x,y) (Where the pixel (x,y) lies in  $N_8(p)$ ), then the current indices are changed to (x,y) and the pixel (x,y) is marked as visited.
4. If no black pixel is found in  $N_8(p)$ , it implies that a discontinuity is encountered. Now the distance transform is applied to the matrix A and the shortest path to the white pixel from the current pixel 'p' is followed. It is necessary to perform distance transform on image 'A' and not on 'B' because the distance transform computes the Euclidian distance between each pixel and the nearest non-zero pixel. Hence performing Distance transform on matrix B would be useless. The path that leads to a white pixel with least distance is traversed. This is done by following the indices of pixels having minimum values in the distance transform matrix. While following this path, the corresponding pixels of the image 'B' are changed to black and are marked visited.
5. The process is continued until the entire boundary has been traced.
6. The loop is terminate if at least 2/3 of the black pixels are traversed and the current pixel 'p' lies in the vicinity of the start point. This ensures that the loop is not terminated in the starting itself. Thus the entire boundary gets traced.

The result of this step of the algorithm in comparison with the thinned image obtained from the previous step is shown in Fig. 7.



**Fig 7:** Comparison of the thinned image (result of the last scan) with the traced image

We can see that all the discontinuities have been linked. And now the region filling algorithm can be applied to the image.

### 2.3 Region Filling

The image obtained after linking the discontinuities (Fig. 7) is first dilated (see Fig. 8) and the Boundary Fill region filling algorithm is then applied to the dilated image (Fig. 9). Dilating the image results in thickening of the contour which eases the process of region filling.

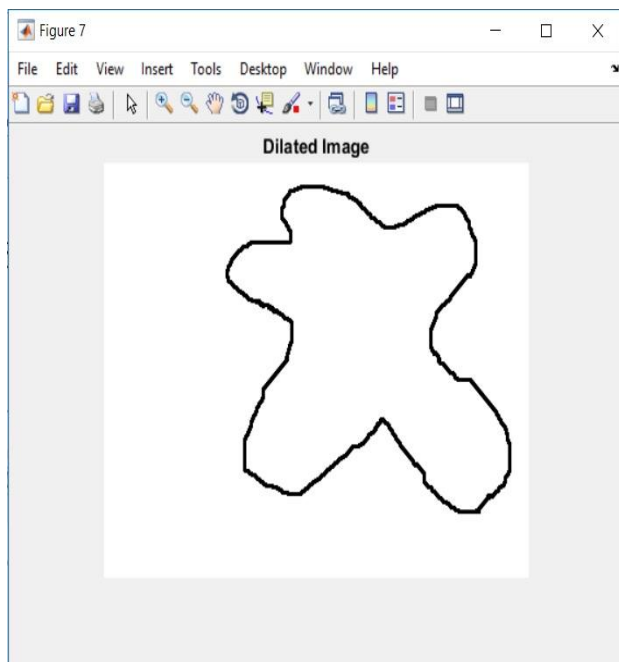


Fig 8: Dilated Boundary

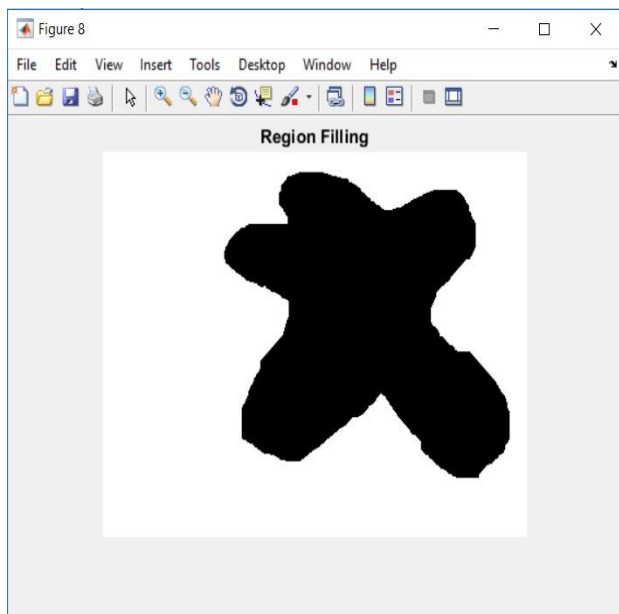


Fig 9: Result of region filling

Thus we have successfully filled the given contour. This would not have been possible by directly applying any of the region filling algorithms because it would result in blackening of the entire image along with the contour.

### 3. RESULTS

Results of some of the test images having different characteristics on which the proposed algorithm was applied along with the corresponding input images are shown in the following Figs 10 - 12.

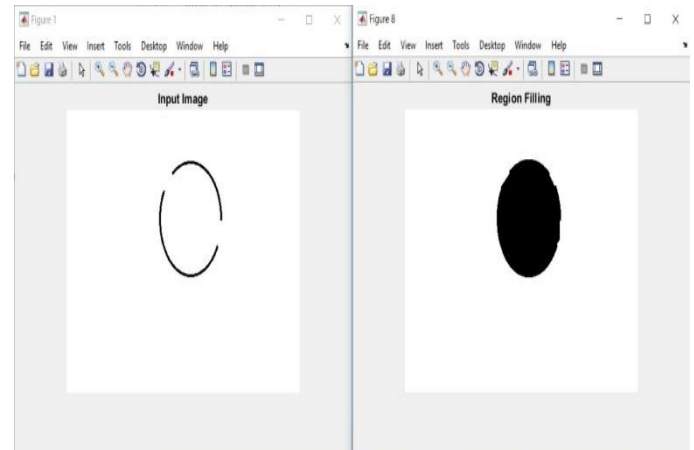


Fig 10: Test Case 1

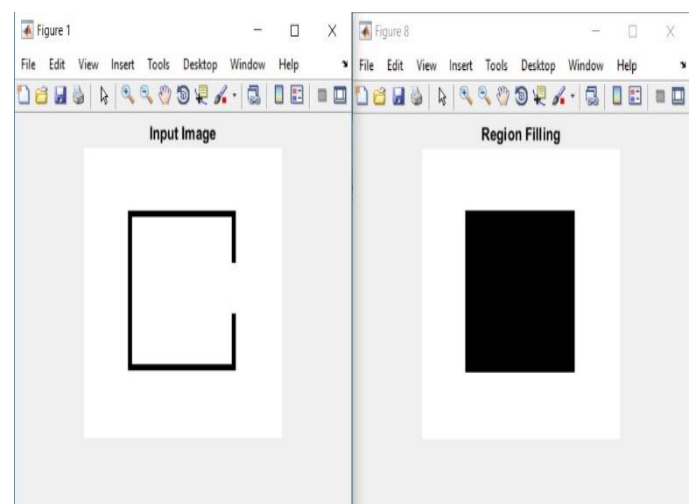


Fig 11: Test Case 2

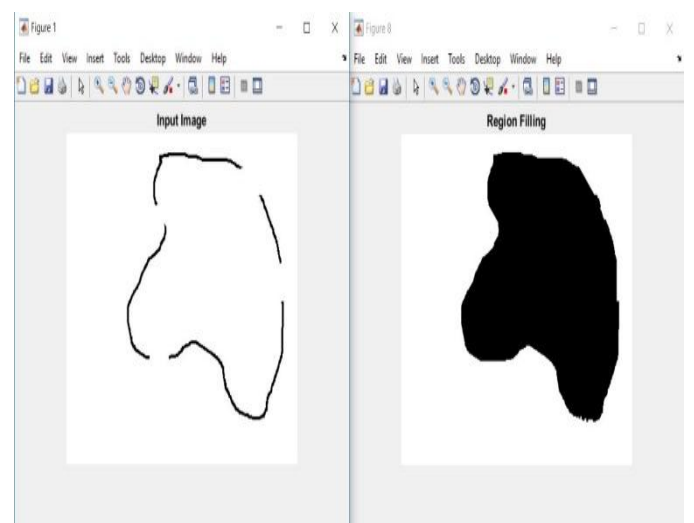


Fig 12: Test Case 2

It can be observed that irrespective of the discontinuities along the boundary, the algorithm has successfully filled the regions after linking the imperfections in the contour.

#### 4. EXECUTION TIME OF THE PROPOSED ALGORITHM

The execution time of the algorithm was found by starting the stopwatch at the beginning of the MATLAB code using “tic” and reading the elapsed time at the end of the algorithm by using “toc”.

CPU Clock frequency = 2.50 GHz

Image Size = 512 x 512

- The average time required to complete the boundary (i.e. excluding region filling) was found to be 250 ms.
- The average time taken to perform the all the operations was on the image was found to be 350 ms.

It is important to note that the entire algorithm was divided into separate functions which were frequently called throughout the execution of the algorithm. This was done with the intention of improving the understandability of the code and also to ease the task of debugging. Hence the execution time mentioned above includes the delay due to the function call and return. The execution time can be improved by avoiding frequent function calls (i.e. by simply merging all the functions) and also by using faster and efficient region filling algorithms.

#### 5. DISCUSSION AND CONCLUSION

The main problem in region filling is that the boundary of the contour has to be continuous. If minor discontinuities are present, then the problem can be overcome by applying simple image processing techniques. However, this is not applicable for larger discontinuities. Applying region filling algorithms to such boundaries will not produce the desired result.

The algorithm described above first reduces the thickness of the contour to 1 pixel. This is very crucial because, if the thickness is more than 1 pixel then it would be very difficult to traverse along the boundary of the contour and more number of iterations would be wasted in evaluating redundant pixels which lie in the same row and column. This would also increase the execution time of the algorithm.

Region filling is one of the basic morphological operation that has a number of practical applications. But, in practical scenarios we cannot expect the scanned images or images captured by a camera to have all the details of the object, there may be imperfections in the image due to which region filling may not be possible. In such cases the proposed algorithm gets the job done.

Some of the applications where the algorithm can be implemented are as follows:

- Medical Image Processing: Often region filling is used in medical image processing to highlight the tumors, cancerous cells that have been captured by the camera so that the affected tissues can be easily observed.
- Contour detection: Contour detection is nothing but determining the shape of the contour based on the number of edges detected. If any of the edges of the contour are broken then the suggested algorithm can be applied to link the discontinuities and the resultant image can be used for the detection process. Contour detection and pattern recognition are fundamental steps of computer vision.

#### 6. FURTHER IMPROVEMENTS

The execution speed of the algorithm can be improved by using a better region filling technique with it. As discussed before there are a number of algorithms that fill the region faster but require perfect contour, which can be obtained by applying the algorithm discussed in the previous sections.

The algorithm can be further improved by scanning the boundary from the interior of the shape instead of scanning from all the four sides. Scanning along different directions from the exterior of the contour may consume some extra time but provides good results, as more details of the contour can be captured by viewing the contour from different directions.

#### REFERENCES

- [1]. Brya N D .Agkland , Member, Ieee, An D Nei L H . Weste, “The Edge Flag Algorithm—A Fill Method for Raster Scan Displays,” Ieee Transactions On Computers, Vol. C-30, No. 1, January 1981
- [2]. T. Pavlidis, "Filling Algorithms For Raster Graphics," Dep. Elec. Eng. and Comput. Sci., Princeton Univ., Princeton, NJ, Rep. 238, Jan. 1978.
- [3]. XunGen Li, LiangMen Huang, “New Region Filling Algorithm Based on Chain Codes Description,” Hangzhou Dianzi University, 3rd International Congress on Image and Signal Processin, 2010.
- [4]. Ren mingwu, Yanjinyu, Sunhan. “A New contour Filling Algorithms Based on Chain Codes Description,”Journal of Image and Graphics-2001.
- [5]. W. D. Little and R. Heuft, "An area shading graphics display system," IEEE Trans. Comput. vol. C-28, pp. 528-531, July 1979. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
- [6]. Freeman H, “The Encoding of Arbitrary Geometric Configuration,” 1961.
- [7]. A. R. Smith, "Paint," NY Inst. Tech., Comput. Graphics Lab., Tech. Memo 7, July 1978.
- [8]. Roger D. Hersch, “Descriptive Contour Filling of Partly Degenated Shapes”, Swiss Federal Institute of Technology, Lausanne.
- [9]. Roman Khudeev, “A New Flood-Fill Algorithm for Closed Contour,” Alparysoft R&D, Tomsk, Russia, Conference on Control and Communication – 2005.

**BIOGRAPHIE**

**Arshad Javeed** – B.E. Student at Sri Jayachamarajendra College of Engineering, Karnataka, India. His areas of interest include Software Engineering, Image Processing, and Computer Vision.