# A Novel Algorithm for Edge Thinning in Binary Digital Images

Dibyendu Ghoshal and Biswajit Paul
Department of Electronics and Communication Engineering
National Institute of Technology, Agartala
Tripura (West) 799055. India
Email: tukumw@gmail.com
Telephone: +91-381-2346630, Fax: +91-381-2346360

***Abstract*—An algorithm has been proposed edge thinning in binary digital images. Normally in practical situations, the edges are produced more than one pixel thick and thus requiring thinning. The double pixel thick edges are first required to be searched and the present algorithm has been prepared with that aim. Subsequently a pointing method has been followed. The edge thinning process starts by searching in horizontal direction as a 1x2 elements, in vertical direction as a 2x1 elements and check the immediate preceding and next pixel. The presence of double edge would be located when there are two consecutive higher intensity pixels will be sandwiched between two low intensity pixels. If the searching in this fashion fails, the method points to 3x4 elements (for horizontal) and 4x3 elements (for vertical) by keeping first (1x2 and 2x1) two elements in the center. Then the algorithm checks immediate top and bottom elements. If no double edge is found in this process, the algorithm checks the upper and lower diagonal elements. The location of the edge, after thinning will be along the direction having higher congregation of high intensity pixels.**

***Keywords—edge thinning; double edge; higher congregation.***

## I. INTRODUCTION

Image segmentation has become an important part in digital image processing with a view to subsequent higher level processing like machine vision [1], image analysis and registration [2] as well as computer graphics [3]. Edge detection has been found to occupy the basic place in the segmentation process of digital images. Edge detection can be accomplished with the help of sharpening spatial filters with specific dimension and weights. Out of these, Roberts, Prewitt, Sobel, Canny, Laplacian, LoG etc.are notable [4 -7]. An ideal edge should comprise one pixel thickness with high intensity value and the high intensity edge is to be surrounded by two lower intensity lines. It has been found that after the execution of edge detection process, edges have two pixels width. An edge have two pixels thickness gives rise to lot of problem regarding the exact location of the edge, the actual difference between the interstices of the adjacent pixels. Double edge may also give rise to handling of more bits of high intensity. Thus edges with more than one pixel need be thinned. There have been a large number of studies regarding edge thinning operation [8]. Lam et al [9] followed an iterative process which was carried out in parallel and serial modes.

The strength of the present algorithm lies in its simplicity of operation. The present algorithm carries out the dual job simultaneously, viz. to locate the edge with double pixels and to find out the location and nature of the thinned edge having width of one pixel. Moreover the algorithm equally holds good when the detected edges have width more than two pixels like three or four because the basic mode of operation and subsequent and subsequent executions of the present algorithm remain the same. Finally, the proposed algorithm does not require to plunge into the in-depth of various branches of applied mathematics.

Although various researchers have carried out a good numbers of studies [11-16] on edge thinning process in the field of digital image processing, their studies mainly centered around complicated mathematical calculation which towards greater complicacy in algorithm generation as well as execution of programs, no paper based on simple algorithm as the present one is focused in available published and online literature.

## II. PROPOSED ALGORITHM

Proposed algorithm can be divided into two parts namely horizontal searching and vertical searching. After horizontal searching the vertical searching has to be done. The section A and B shows the pseudo code of the proposed method and description.

### *A. Pseduo Code horizontal searching:*

M is the number of rows and N is the number of columns.

1. For i = 2: M-1;

   For j = 2: N-1;
2. Select two 1-D elements as f(i, j:j+1);

   If f(i, j) == 1 and f(i, j+1) == 1

   Continue

   Else go to step 1;
3. Perform checking f(i, j) with f(i, j-1) and f(i, j+1) with f(i, j+2)

   If f(i, j-1) == 1 or f(i, j+ 2) == 1

 ISBN: 978-81-923777-9-7

go to step1;

break ;

Else continue

4. Point to 3x4 element keeping first 1-D element in the centre as f(i-1: i+1, j-1:j+2)

   If (f(i- 1, j) == 1 and f(i+ 1, j) == 1) and (f(i-1, j+1) ~= 1 or f(I +1, j+1) ~= 1)

   Edge is placed on f(i, j) and f(i, j+1) is set to zero.

   Else if (f(i- 1, j) ~= 1 or f(i+ 1, j) ~= 1) and (f(i-1, j+1) == 1 and f(I +1, j+1) == 1)

   Edge is placed on f(i, j+1) and f(i, j) is set to zero.

   Else break;

5. Check for the diagonal element;

   If (f(i- 1, j- 1) == 1 and f(i+ 1, j-1) == 1) and (f(i-1, j+2) ~= 1 or f(i +1, j+2) ~= 1)

   Edge is placed on f(i, j) and f(i, j+1) is set to zero.

   Else if (f(i- 1, j- 1) ~= 1 or f(i+ 1, j-1) ~= 1) and (f(i-1, j+2) == 1 and f(i +1, j+2) == 1)

   Edge is placed on f(i, j+1) and f(i, j) is set to zero.

6. go to step 1 and continue upto i= M-1 and j= N-1;

The details of the above code are described below:

Step 1: In horizontal searching the proposed algorithm searches two horizontal elements on image matrix and the choice starts from f(2, 2) and f(2, 3) element as shown in Fig.1.

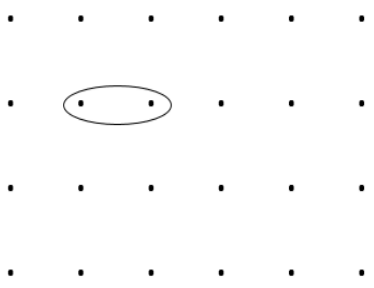


Fig. 1. The elements f(2, 2) and f(2, 3) have been selected for starting elements.

Step 2: The immediate next and previous elements (pixels) along the horizontal direction are compared. If only one of the two adjacent elements or both elements are found to be white i.e. 1, then the process would return to the step 1 to search for another adjacent high intensity pair of pixel along horizontal direction and the present step is repeated in case the adjacent pixels are with low and high intensity then there will be an edge. But this edge will consist of two high intensity pixels already there was a pixel with high intensity in the left.

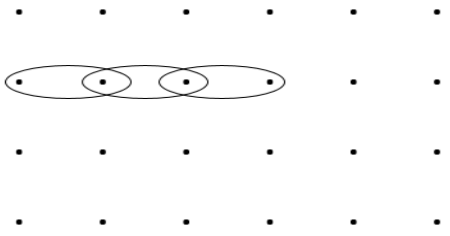


Fig. 2. The elements f(2, 1) and f(2, 4) have been selected as previous and next element.

During the execution of step 2, if the next (nearest) pixel happens to be one (white), then the entire horizontal line will constitute one line with all having high intensity level (i.e. white line in binary image) but the situation cannot make any prediction about the other surrounding pixels which in tern cannot detect any edge. But when the adjacent pixel is zero, then it can be inferred that the initial two neighboring pixel form the double edge as shown in Fig. 3.

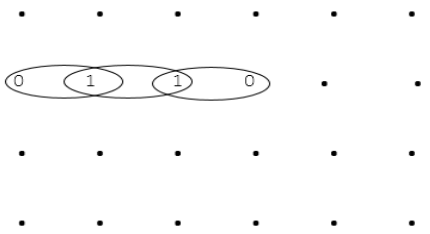



Fig. 3. Double edge is formed for the value of the adjacent pixel is zero.

Step 3: When the double edge is determined by the step 2, then it is required to explore nearest pixels of the original pair along the vertical direction as shown in Fig. 4.

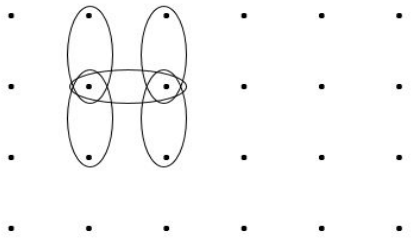


Fig. 4. Double edge elements are compared with the top and bottom elements.

Here both sides of the original pair will be explored. If it is found that along one side, the number of pixels having value one is more than the other, the edge exists along that direction because along the other direction, number of ones is less and it indicates that it is not a continuous line. In this the vertical pixel intensities are compared with the original horizontal pair of pixels having one value. If it is found that along any one side, both the upper and lower pixels have value one, then then it indicates an edge with one pixel thickness. The same would be true for the other side of the original left pixel (say along right side). In case, along both the sides of the original pair have high intensity value pixel, this would indicate two edge


ISBN: 978-81-923777-9-7

line along both the sides with one pixel thickness. It implies two adjacent edges having thickness of one pixel i.e. it will turn out to be an edge with double pixel width. In this case, the pointer will automatically follow the next step.

Step 4: When the step 3 fails to thin the double edge, then the present algorithm searches the nearest diagonal pixels of both the original pixel pair having one value as shown in Fig.5.

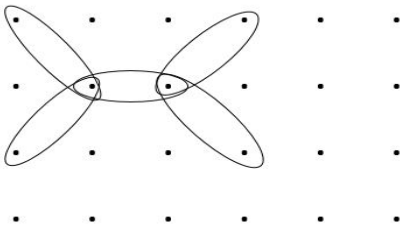


Fig. 5. Double edge elements are compared with the upper and lower diagonal elements.

In this case also, the number of more one value pixel will be explored and if is found that the number of white pixels are more (including the adjacent diagonals) on the right side than that of the left side, then edge exists along the right side because here it will form one white continuous line with the thickness of one white pixel when the left side (having lower number of pixel white value) of original pixel will be replaced by black pixel thereby accomplishing the edge thinning for horizontal searching.

*B. Pseduo code for Vertical Searching:*

M is the number of rows and N is the number of columns.

1. For i = 2: M-1;
   For j = 2: N-1;
2. Select two 1-D elements as f(i: i+1, j);
   If f(i, j) == 1 and f(i+1, j) == 1
   Continue
   Else go to step 1;
3. Perform checking f(i, j) with f(i-1, j) and f(i+1, j) with f(i+2, j)
   If f(i-1, j) == 1 or f(i+2, j) == 1
   go to step1;
   Break ;
   Else
   Continue
4. Point to 4x3 element keeping first 1-D element in the centre as f(i-1: i+2, j-1:j+1)
   If (f( i, j-1) == 1 and f(i, j+1) == 1) and (f(i+1, j-1) ~= 1 or f(i +1, j+1) ~= 1)
   Edge is placed on f(i, j) and f(i+1, j) is set to zero.
   Else if (f(i, j-1) ~= 1 or f(i, j+1) ~= 1) and (f(i+1, j-1) == 1 and f(Ii+1, j+1) == 1)
   Edge is placed on f(i+1, j) and f(i, j) is set to zero.
   Else Break;
5. Check for the diagonal element;
   If (f(i- 1, j- 1) == 1 and f(i+ 2, j-1) == 1) and (f(i-1, j+1) ~= 1 or f(i +2, j+1) ~= 1)
   Edge is placed on f(i, j) and f(i+1, j) is set to zero.
   Else if (f(i- 1, j- 1) ~= 1 or f(i+ 2, j-1) ~= 1) and (f(i-1, j+1) == 1 and f(i +2, j+1) == 1)
   Edge is placed on f(i+1, j) and f(i, j) is set to zero.
6. go to step 1 and continue upto for i= M-1 and j= N-1;

The details of the above code are described below:

Vertical searching is similar to the horizontal searching, it can be viewed as dual of horizontal searching.

Step 1: In vertical searching the proposed algorithm searches two vertical elements on image matrix and the choice starts from f(2, 2) and f(3, 2) element as shown in Fig. 6.

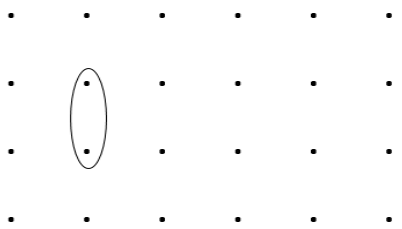


Fig. 6. The elements f(2, 2) and f(3, 2) have been selected for starting elements in vertical searching.

Step 2: The immediate top and bottom elements (pixels) along the vertical direction are compared. If only one of two adjacent elements or both elements are found to be white i.e. 1, then the process would return to the step 1 to search for another adjacent high intensity pair of pixel along vertical direction.

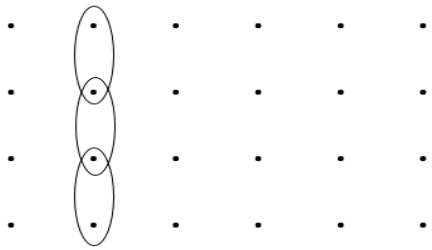


Fig. 7. The elements f(1, 2) and f(4, 3) have been selected as top and bottom element.


ISBN: 978-81-923777-9-7

Step 3: When the double edge is determined by the step 2, then it is required to explore nearest pixels of the original pair along the horizontal direction as shown in Fig. 8.

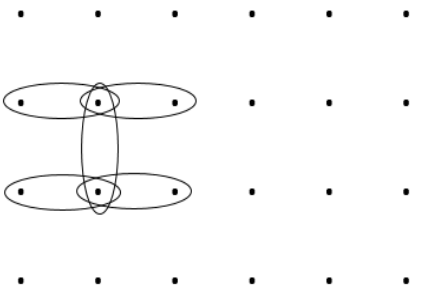


Fig. 8. Double edge elements are compared with the immediate next and previous elements.

If it is found that along one side, the number of pixels having value one is more than the other, the edge exists along that direction.

Step 4: When the step 3 fails to thin the double edge, then the present algorithm searches the nearest diagonal pixels of both the original pixel pair having one value as shown in Fig.9.

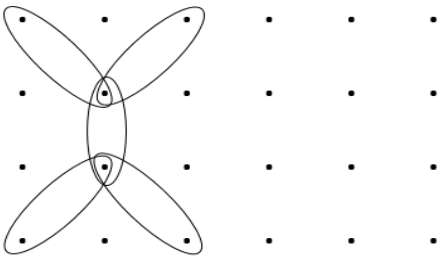


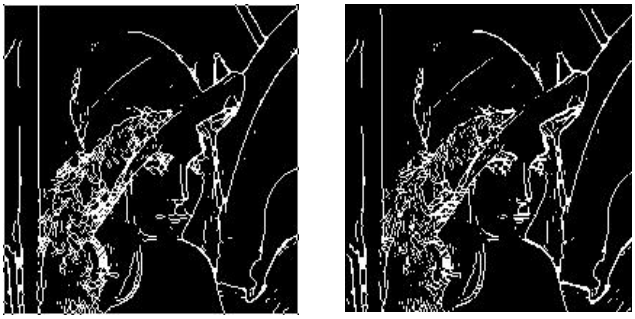
Fig. 9. Double edge elements are compared with the upper and lower diagonal elements.

The final location of the thinned edge line will be along the white line.

## III. Results and Discussion

The proposed method has been developed using MATLAB 2009b and employed on a edge detected image of lena of size 204x204 using Compass operator [10] as shown in Fig.10.

(a) (b)



(c) (d)

Fig. 10. (a) Original image, (b) edge detected using compass operator, (c) edge thinning using [3], (d) edge thinning using proposed method.

The main advantage of the proposed algorithm is that the entire image is converted by an exploring window whose dimension may have any convenient value. The existence of the double edge and removal of the same is done in step when the window is moved from left column to rightmost column of the digital image. At the end of the operation, the window traces back to the left most column side and restart its operation and in that way the window covers the entire image, leaving no pixel unexplored. The existence of the double edge will be definitely checked and verified if there any such exists. At the same time, the edge is thinned by simply replacing one white pixel to a zero by programming technique.

The proposed algorithm does not require any post processing such as edge linking. This will execution faster and computational limit will be expectedly less. There is also no need for iteration for the comparison between the pixels.

## IV. Conclusion

The proposed algorithm is simple, expectedly computationally faster and devoid of any post processing like edge linking [17]. Furthermore, this algorithm does not require to handle any morphological operation like to find the skeleton of the image etc.

The drawback of this proposed algorithm is that it does not robust to impulse noise [10] whose presence may alter any white pixel to the black value and vice versa.


## V. Acknowledgment

Authors are thankful to Prof. P. K. Bose, Director National Institute of technology, Agartala for his constant kind inspiration and support.


## VI. Dedication

One of the authors (Dibuyendu Ghoshal) dedicates the entire study to the loveliest And loving memory of his only one and younger sister Kumari Sumita Ghoshal who herself was a gem of the scholars, a symbol of wisdom and art, peerless beauty and simplicity, unfathomable knowledge and generosity.


ISBN: 978-81-923777-9-7

## REFERENCES


[1] I. Sobel, "Camera models and machine perception," Stanford University, Stanford AI MEMO 121, 1970.

[2] W. Pratt, Digital Image Processing. 1ohn Wiley & Sons, 1991.

[3] R. Gonzalez and R. Woods, Digital Image Processing. Prentice Hall, 2008.

[4] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679 -698, nov. 1986.

[5] E. Nezhadarya, and R. K. Ward, "A New Scheme for Robust Gradient Vector Estimation in Color Images," IEEE Transactions on Image Processing, vol. 20, pp. 2211 – 2220, 2011.

[6] X. Wang, "Laplacian Operator-Based Edge Detectors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 886 – 890, 2007.

[7] S. Pande, V. S. Bhadouria, and D. Ghoshal, "A study on edge marking scheme of various standard edge detectors," International Journal of Computer Applications, vol. 44, pp. 33- 37, 2012.

[8] J. –D. Dessimoz, "Specialized edge-trackers for contour extraction and line- thinning," Signal Processing, vol. 2, no. 1, pp. 71-73, 1980.

[9] L. Lam, S. -W. Lee, and C. Suen, "Thinning methodologies-a comprehensive survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, pp. 869 -885, sep 1992.

[10] A. K, Jain, Fundamentals of Digital Image Processing. Prentice Hall of India, 2001.

[11] C. Genming, and Y. Boazong, "A New Edge Detector with Thinning and Noise Resisting Abilities," Journal of Electronics, vol. 6, No. 4, pp. 314-319, oct. 1989.

[12] R. M. Noorullah, and A. Damodaram, "Innovative Thinning And Gradient Algorithm For Edge Field And Categorization Skeleton Analysis Of Binary And Grey Tone Images," Journal Of Theoretical And Applied Information Technology, vol. 5 ,pp.75-80, 2009.

[13] T. Y. Zang, and C. Y. Suen,"A Fast Parallel Algorithm for Thinning Digital Patterns," Comm. ACM, vol. 27, no. 3, pp. 236-239, 1984.

[14] T. Suzuki, and S. Mori, "A Thinning Method Based on Cell Structure," Int. Workshop Forntier Handwriting Regn, pp. 715-719, 1990.

[15] R. M. K. Singha, and C. J. Ammann, "Comments on Fast Thinning Algorithm for Binary Images," Image Vision Comput., vol. 4, no. 1, pp. 57-58, 1986.

[16] "A Thinning Algorithm by Contour Generation," Comm. ACM, vol. 31, no. 11, pp. 1314-1324, 1988.

[17] Z. Wang, And H. Zhang, "Edge Linking Using Geodesic Distance and Neighborhood Information," International Conference on Advanced Intelligent Mechatronics, pp. 151- 155, July 2008.



ISBN: 978-81-923777-9-7