Improving Morphology Operation for 2D Hole Filling Algorithm

Mokhtar M. Hasan mmwaeli@gmail.com
Computer Science Department/ Faculty of Science
Banaras Hindu University
Varanasi, 221005, India

Pramod K. Mishra pkmisra@gmail.com
Computer Science Department/ Faculty of Science
Banaras Hindu University
Varanasi, 221005, India

Abstract

Object detection may result with some noises, the correct detecting of such object plays a major role for later recognition steps, the interior noise of the object must be removed, the morphological operations are used successfully for this purpose, these morphological operations are applied for 2D holes filling using dilation operation, in this paper we have enhanced this algorithm to get better and faster version that will reduce the processing time dramatically by using a dynamic marker, we have applied two kind of markers with different structuring elements but same size which is 3x3 and those markers are used according to the structure of the sub-window of the object, the processing time is reduced, and our algorithm reduces this time approximately to one third, the results also enhanced since there are some cases missed by the extant version of morphological operations holes filling algorithm.

Keywords: Morphological Operation, Dilation Morphological, Object Filling, 2D Object Filling, Object Holes, Noise Removal, Scanning 2D Objects.

1. INTRODUCTION

We must start with the formal definition of the hole, cavity, and concavity, the hole has been defined from different views but all referring to the same common ground, we can define the hole as "a background region that is a subset of a object", since the hole region should be a subset of their object that belongs to, the Gonzalez and Woods definition in [1] was verbatim *"background region surrounded by a connected border of foreground pixels"*, the other definition that found in Oxford Dictionary [2] was "*a hollow space in something solid or in the surface of something*", in all of the mentioned definitions; we have to exclude the cup handle from being a hole since it is part of the cup [3] or the area that lies in the middle of a doughnut [3] as well, the 3D definition of the hole can be taken from [3] which is "*break in a surface mesh".*

The cavity and concavity are dedicated for 3D objects since these terms has a more understandable meaning, the Oxford Dictionary [2] for this term verbatim is "*a hole or empty space inside something solid*", so, we have moved to the space view of the hole, the other definition [4] is the cavity is the existence of a hollow [2] in an object, or verbatim from [4] "*a bounded connected component of the background*", while the concavity is considered to be a contour with concave shape [2].

Object scanning may produce some missing portions or missing regions even pixels of the scanned object due to and hence, we address the hole filling algorithms to close up and fill out this missing information.

There are many usages of the holes filling algorithms, the preprocessing steps may add some extra holes [5] to the processed object since the segmentation operation is not always perfect, so, applying the hole filling will eliminate the original holes and the new added holes as well [5]. Other usage is for Cultural Heritage [6] which has several reasons, the authors in [6] summarize those reasons by the sampling accuracy [6], speed gained from scanning technique [6], and acquiring these object data without touching the object [6] since some objects are very fragile due to their ages.

Other factors that impact upon the producing of the holes are occlusion [5], low reflectance [5] or the original scanned object has some missing portions [5].

Furthermore, in order to produce a best model that fits the original object [7] we need to remove any noises that classified as object region or foreground region.

In this paper we have modify the morphology operation for object filling and we gain a more faster and more enhanced results since there are some missing cases by the original algorithm. This paper is organized as follows: section 2 lists some related work associated before, part 3 gives a review for the meaning of the morphological operations, part 4 shows the other algorithms applied in this field as well as the original version of the algorithm that we have modified, part 5 shows our algorithm, part 6 explains the impact of the selection of the structuring elements that we have made to overcome some shortcoming issues that arose from the original version of the extant algorithm, part 7 shows the experimental results, part 8 shows the speed factor comparisons between the old version and our version, and finally, part 9 gives the conclution.

2. RELATED WORK

Gonzalez and Woods [1] presented two different algorithms for object hole filling, these algorithm use the morphological dilation operation for such purpose, the first algorithm is restricted since an initial pixel within the object hole must be specified, which is more difficult in real time automatic applications that needs the filling to be done completely without human interference.

The other algorithm presented by them was applied by dilation algorithm as well but this time no initial pixel needed to be located. They started with an initial marker which is the complement of the border of the original input image and black pixels elsewhere, and their 3x3 structuring element (SE for shorthand) was all ones, and by applying the algorithm, the border comes and closing the objects as every epoch of the algorithm until there is no more changes in the resulted marker, and by anding logical operation between the complement of the marker and the original image matrix, they got the image holes matrix and they got the filled image by oring logical operation between the last two matrices.

The other approach was applied by us in [7] which depends on finding a connected region around the background pixel to be recognized as hole pixel, any initials or prior assumption have not required, they convolved a diamond filter with every background pixel and a roadmap is created, then they trace this road map to find a connected path which means the background pixel has a connected region surrounds it, if such path existed; then this hole pixel as foreground pixel is considered, otherwise, background pixel is considered .

3. MORPHOLOGICAL OPERATION: REVIEW

The term of morphology came from the biology branch [1] which means the study of the structure of the animals and plants [1], the same term can applied to image objects for studying the structure of these objects in the image [1], there are two basic and important operations which are the dilation and erosion [1], the Oxford Definition [2] for dilation is "*to become or to make something larger, wider or more open*", and the Oxford Definition [2] for erosion is "*to gradually destroy the surface of something through the action of wind, rain, etc*", and in the image erosion , the object is eroded by the action of filters which is called the SE that has a reference point that dominate the matching with the object window since we have no wind nor rain here.

3.1 Morphological Operation Notations

Let A ∈ Z^2, which is the 2D space of the (x, y), and also let B ∈ Z^2 be the SE which controls the structure of the morphological operations, then for binary images, dilation and erosion can be defined as (1) and (2) respectively:

$$A \oplus B = \{ z \mid z = a \,.\, b \,,\, \exists\, a \in A \;\&\; b \in B \} \tag{1}$$

$$A \ominus B = \{ z \mid z = a \,.\, b \,,\, \forall\, a \in A \;\&\; b \in B \} \tag{2}$$

Where ⊕ and ⊖ represent the dilation and erosion operations respectively, and **.** (dot) represents the logical anding operation, Equation (1) says that the output of the dilation operation is set to reference point if there is any hit between object A and SE B, this reference point represents the output value if such hit existed, and for Equation (2); all SE must be included inside the object A to produce the reference point as an output which means fit.

The other term is the reconstruction which may exploit the dilation or erosion to get its process done; we will focus on dilation since it is used in this paper, the formulation for this operation as in (3) which taken from [1]:

$$R_A^D(F) = D_A^{(k)}\,(F) \tag{3}$$

Where R denotes for reconstruction operation, D for dilation, A is the mask image, F is the marker and k represents the number of repeated times until (4) holds:

$$F^k = F^{(k+1)} \tag{4}$$

We will define no more operations since it is out of the scope of this paper, any more definition can be referred to [1,8].

4. MORPHOLOGICAL OPERATIONS FOR HOLES FILLING

The formulation of the two morphology filling operation can be formulated as follow:

4.1 Hole-Pixel Initial Algorithm (HPIA)

Consider O = { o_0, o_1, .., o_k | o_i represents the (x, y) pixels of object i, k represents the number of objects in image }, then the initial for this algorithm is a vector P where in (5) and (6) :

$$P=\{I_i \mid i=1, 2, \dots k\} \tag{5}$$

$$I_i = \{(x_j, y_j)\ ,\quad j \in \text{number of holes in object } i\ \} \tag{6}$$

This required having starting points that belongs for each hole in each object of the image, which is very difficult since the real time application for such algorithm cannot go along with this since it requires the human interaction in a very essential step which is all the latter processing depends on. However, the The equation for this algorithm is in (7) taken from [1] which closes the holes of the object after certain number of epochs which means until there is no change in X_k:

$$X_k = (X_{k-1} \oplus B\ ) \cap A^c\ , \text{for } k = 1, 2, 3, \dots \tag{7}$$

Where X_0 is the marker with initial points I_i , B is the 3x3 SE with zeros at the corners and ones elsewhere, and A is the original input binary image.

4.2 Border Image Initial Algorithm (BIIA)

The algorithm is more efficient for hole filling and the initial value for this algorithm as in (8) which is taken from [1] since it is their algorithm:

$$F(x,y) = \begin{cases} 1 - A(x,y)\ , if\ (x,y) is\ a\ border\ pixel\ of\ A \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \tag{8}$$

Where F is the marker, after that, the algorithm starts the following iteration and repeats (9) (taken from [1]) until there is no more change in the marker:

$$H = [R^{D}_{A^c}\ (F)]^c \tag{9}$$

Where H denotes the final image output. This algorithm also suffers from the speed factor since it starts at the border and after that it will go and expand one pixel at a time from border, many epochs required to find H and this depends on the image dimensions, for a rectangular image with 200x200 pixel, it required maximum of 100 epochs; and in case of the input image has no object the algorithm will continue also wasting the time, and in case of filling the letters of image document, we will notice that this algorithm is not working properly since there are some missing issues.

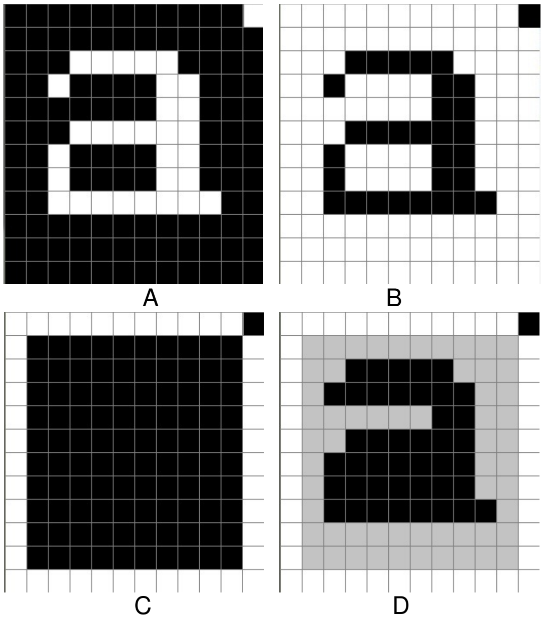
5. OUR SUGGESTED ALGORITHM

We will call our algorithm as Speed Border Image Initial Algorithm (SBIIA) for shorthand. The marker F that used in the previous algorithm had starting points from at the border, if we can increase the number of starting points more than just border points; the dilation operation speed will be increased, we have two ideas in this situation, first idea is the distribution of the random points all over the F, but in this solution; we have to be careful from choosing the random pixel inside the object hole and the distributed points should belong to exterior area of the objects, so, this solution is not working properly as well, the other solution is by scanning the input image row by row starting from the border and filling the corresponding F pixels as a foreground pixels for those are background pixels in A, after that; the job of the dilation will be just to fill the pixels that are between the objects or outside the object in the folded area. Equation (10) shows the initial of marker F.

$$F(x,y) = \begin{cases} 1 \quad if\ there\ is\ a\ path\ from\ border\ to\ (x,y) \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \tag{10}$$

By this initialization, the speed of the BIIA is enhanced to be one third of its present time; the performance also enhanced as well since in case of empty image, our algorithm will iterate just one epoch unlike the BIIA that will waste the time. Figure 1 shows the initial marker by using both of BIIA and SBIIA for a given sample.

The gray cells in Figure 1D are also belonging to marker but to show the extra points unlike Figure 1C, by using this marker as initial, the algorithm will be speeded and the processing time will be dropped down significantly. In other words, the BIIA needs many extra epochs until it stops at the object border which is the aim since the hole area is the one that come from the anding operation between complement of the resulted marker and the complement of the original input image, and less epochs for SBIIA since the marker is more closer to the border of the object.



A: the original image, B: the A^c , C: initial marker by BIIA, D: our initial marker by SBIIA.
FIGURE 1: The Initial Marker as Applied By BIIA and SBIIA.

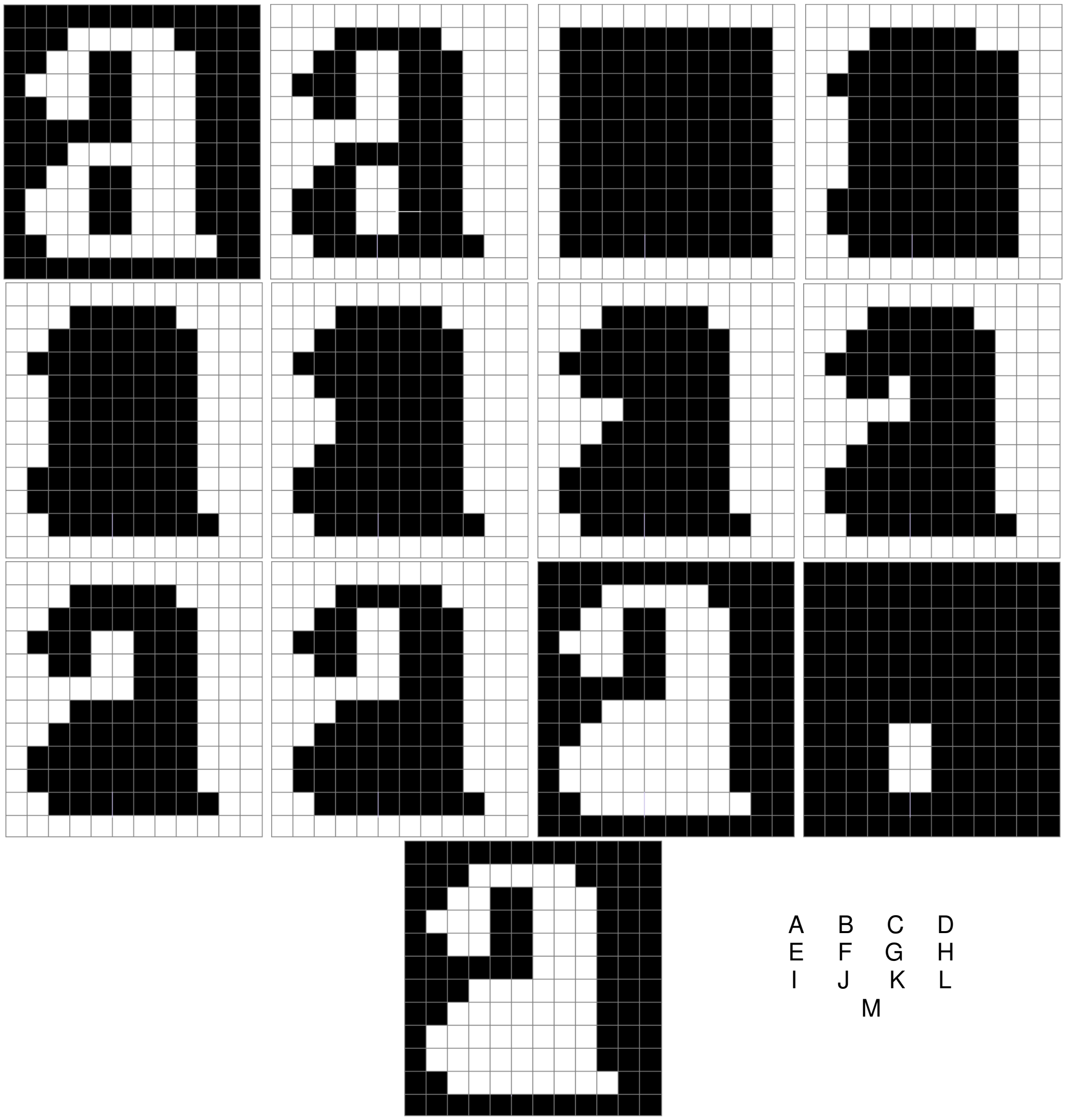
6. STRUCTURING ELEMENT

6.1 SE Operation

Before we penetrate into the selection of the SE, we need to focus some lights on the processing method of the SE, the SE will be matched with the current marker, if there is any hit; the reference point will be produced as an output in the corresponding location of the marker, since we aim to cover all the exterior object area; the input image is first complemented to make the object as background area and background area as a foreground and hence; the searching for the foreground and reconstructing it means the covering for the area that is originally background, after finishing that process, the complement of the marker and the complement of the original input image is anded and the resulting area is the holes which will be ored with the original input image to produce hole-free object as seen by Figure 2.

As seen by Figure 2, the reconstruction operation stops when there will be no change on the produce marker, and this marker is expanded one border pixel at a time, this process continues and stops at a limited number of iteration since the convergence is assured by the expansion process. Furthermore, as noticed by Figure 2 (D-J), this reconstruction operation is obtained by anding the resulted marker at each step with Figure 2B which is the complement of the original input image to ensure this construction operation does need penetrate through the object holes.

The correct selection of the SE plays a crucial role for the successful of the processing operation since it controls the way of expansion or shrinking of the image object, for holes filling, the SE that is used for algorithm in [1] is shown in Figure 3 in which the shaded pixel represents the reference point.



A: the original input image. B: the complement of A. C: initial marker as per BIIA.
D: first step of reconstruction operation as per Eq. (9). E: second step of reconstruction.
F: third step of the reconstruction. G: fourth step of the reconstruction.
H: fifth step of the reconstruction. I: sixth step of the reconstruction.
J: seventh step of the reconstruction. K: the complement of the resulted marker in J.
L: the anding operation between K and B. M: oring operation between L and A.

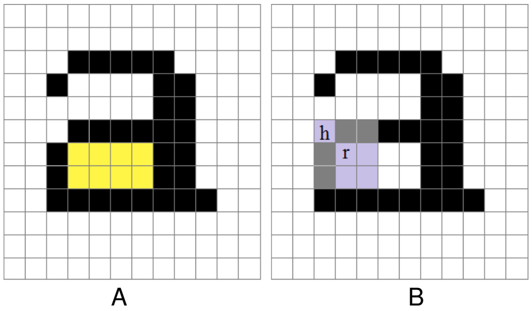
FIGURE 2: The Application of the Filling Operation.

1	1	1
1	1	1
1	1	1

FIGURE 3: The Structuring Element used for Holes Filling.

This SE used here has a disadvantage, which is inability to detect the corner point in case of the edge is thin (one pixel wide), furthermore, this edge is not horizontal nor vertical but diagonal, so, the current version of BIIA continues at this corner edge and expands inside the object which in turn assumes this hole as background since it comes with the trace and the trace should stops at the object border and surrounds the object interior hole as we mentioned hereinabove, in order to give better understanding for this problem; Figure 4 addresses this issue.

As seen by Figure 4B, the SE hits the corner point at (h) and the reference point will be set to background area and the expansion will go through this area that attached the reference point, which is incorrect since the trace should stops at the border and should not proceed inside the object holes(s), hence, the yellow area as seen by Figure 4A will be considered as a background area and the output image will be the same of the input image without change in this case, so, this is the shortage that infiltrated the performance of the BIIA algorithm.



dilation should stop since it is a hole area

A: complement of ‘a’, B: same of (A) during process.

FIGURE 4: Addressing the disadvantage of BIIA.

However, our modification resides in the using of two SE’s, and their use is managed by a specific threshold after anding with 3x3 image sub-window, Figure 5 shows the SE’s used by us.

0	1	0
1	1	1
0	1	0

A: non-threshold SE

1	1	1
1	1	1
1	1	1

B: threshold SE

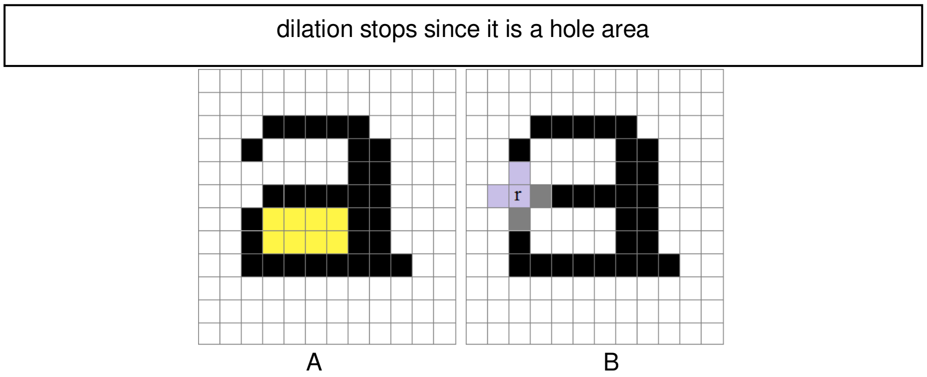
FIGURE 5: Dynamic Structuring Elements Based On Threshold.

6.2 Non-Thresholding SE

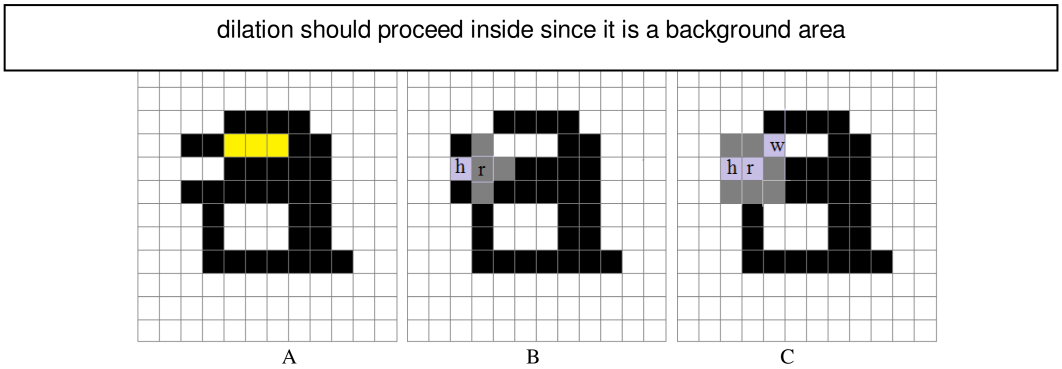
Figure 5A is used by us as a general case for dilation operation, Figure 6 shows the application of our SE on Figure4A and we can notice that the problem addressed in Figure4B in not happening here, since, at this corner; there is not infiltrating for the reference point inside the object hole and this problem that addressed by BIIA is recovered here and the area inside letter ‘a’ will be considered as a hole area and not a background area, since there is no connection between the inner region (yellow region) and outer surrounding one and also since our conditional thresholding SE will not be employed here as we will see in the next section.

6.3 Thresholding SE

The usage of second SE is restricted in case of the summation of overall SE after anding with current sub-window of marker F is equal to a threshold; this threshold is 3, the following Figure 7 illustrates the importance of use such SE in which we can manage the image object having different combinations of corners and non-corners and these corners may be sharp as mentioned in Figure 4 or non-sharp like the left-lower corner at the same latter mentioned figure, normally, we employ the non-thresholding SE in our operation, but, however, if the above stipulation is meet for thresholding SE, this latter SE will be employed to cover the problem of corner points.


dilation stops since it is a hole area

A: complement of ‘a’, B: farthest hit point which still outside of interior hole region and there is no connection between inside and outside regions.
FIGURE 6: Our Remedy for Corner Pixel.

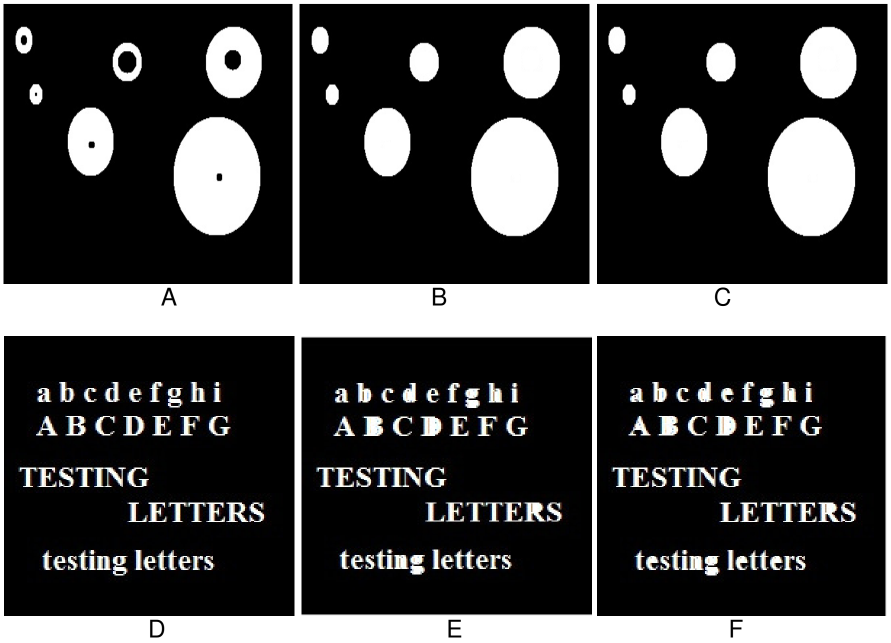

dilation should proceed inside since it is a background area

A: complement of letter ‘a’. B: during process, non-thresholding SE cannot penetrate this region.C: we can expand through yellow region starting at point w.
FIGURE 7: Exception Case For Using Of Threshold SE.

As seen by Figure 7B, the yellow area should be attached to a background region, but this region is attached to hole region if we applied the non-thresholding SE since there is no penetration for reference point inside the yellow area, however, Figure7C shows our implementation for the thresholding SE, this thresholding SE is used in case of there is two hit points (including reference point) and one point inside the hole, so, the total is three points out of the 3x3 structure which matches the state in Figure7C, if we compare this issue with Figure 6, we can see that there are three hit points in latter Figure 7 and there are five hit points in Figure 6 (in case of thresholding SE applied), so, the non-thresholding SE will be enough as a general, experiential results for image document will clear this issue.

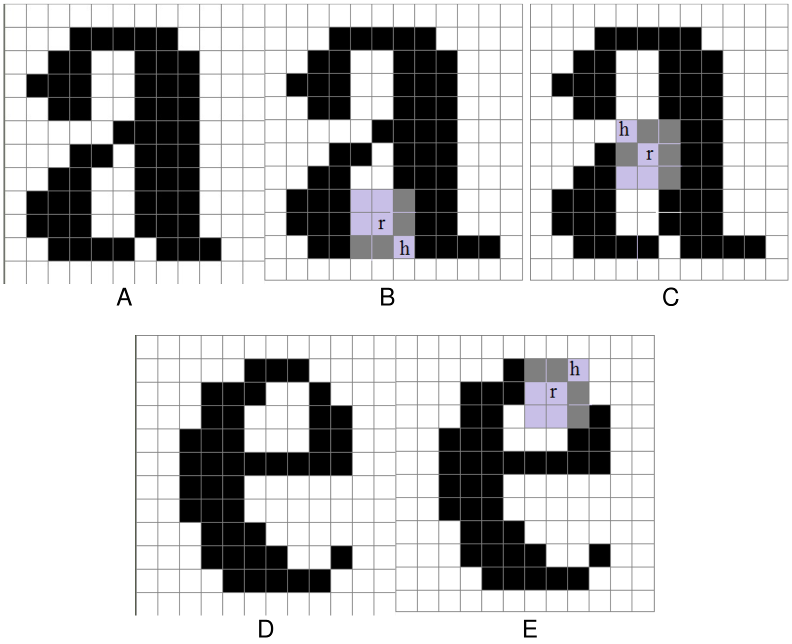
7. EXPERIMENTAL RESULTS

We have tested our suggested algorithm (SBIIA) with many samples and also with document images; we have gained a good and promising filling and high speed ratio compared to BIIA, and also we have listed a time factor comparison in order to reveal SBIIA’s speed. Figure 8 demonstrates the first group of basic sample that will be used as testing objects with their corresponding results after applying BIIA and SBIIA.



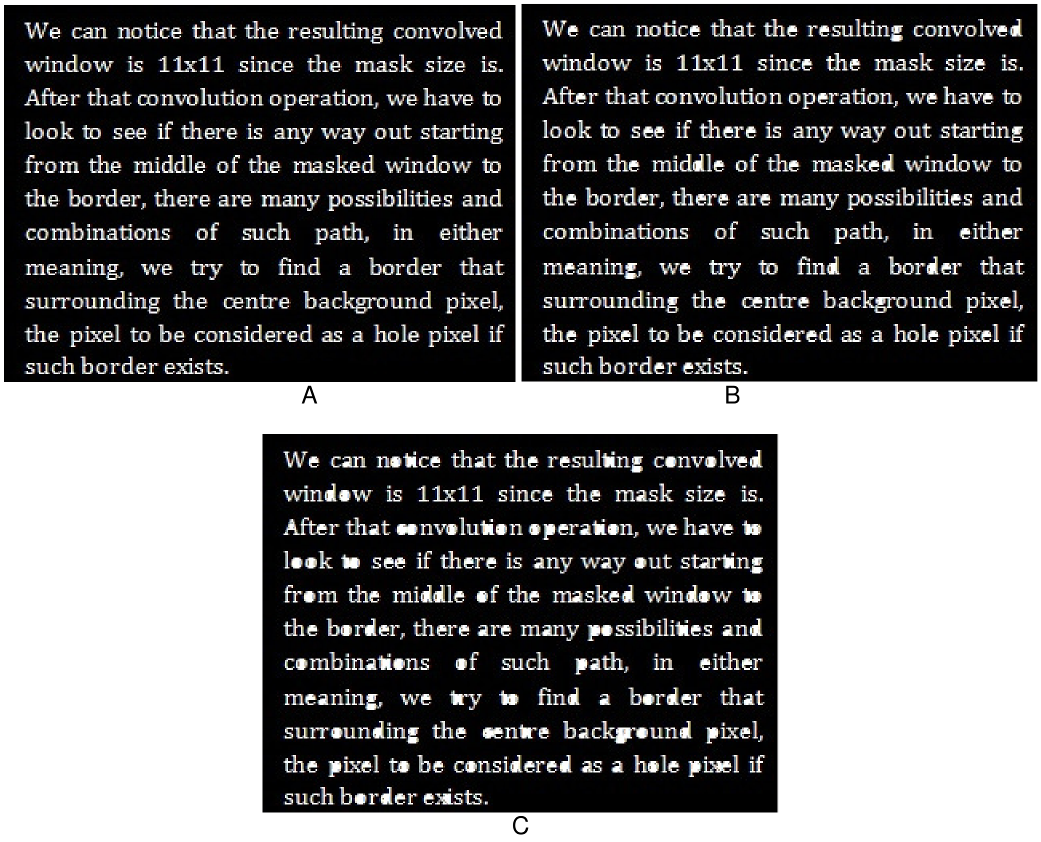
A and D represents the tested samples, B and E is the output of BIIA for each of A and D respectively, C and F is the output of SBIIA for each of A and D respectively.
FIGURE 8: Testing Output Applied on each of BIIA and SBIIA.

As seen by Figure 8E, BIIA is missed 6 letters continuously which are (from top to down, from left to right): a, e, A, e, e, and e, this happened because the problem that we have addressed hereinabove, we made a detailed sketch of letter ‘a’ and ‘e’ from above missed letters in order to give a close look for this issue, see Figure 9.



A and D represent the missed letters, B, C and D represent their process.
FIGURE 9: Illustrating the problem of BIIA.

As seen by Figure 9, those letters are taken from the same image and has been drew with pixel wide image for better understanding, however, at the corner, there will be an entrance inside the hole and hence, will be considered as a background region since it comes with the trace, and at final these letters will be the same of input without any change since, as we mentioned before, the trace should stops at the border and should not come inside the object. This problem has been solved by our suggested method. In Figure 10 we have demonstrated one more text image example.



We can notice that the resulting convolved window is 11x11 since the mask size is. After that convolution operation, we have to look to see if there is any way out starting from the middle of the masked window to the border, there are many possibilities and combinations of such path, in either meaning, we try to find a border that surrounding the centre background pixel, the pixel to be considered as a hole pixel if such border exists.

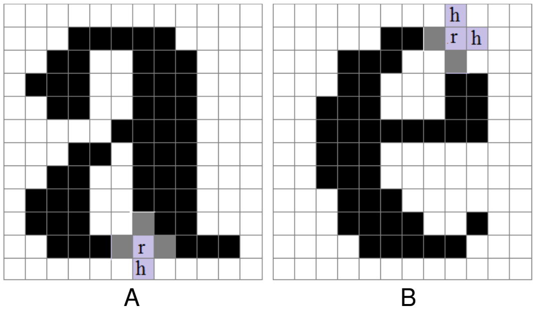
We can notice that the resulting convolved window is 11x11 since the mask size is. After that convolution operation, we have to look to see if there is any way out starting from the middle of the masked window to the border, there are many possibilities and combinations of such path, in either meaning, we try to find a border that surrounding the centre background pixel, the pixel to be considered as a hole pixel if such border exists.

We can notice that the resulting convolved window is 11x11 since the mask size is. After that convolution operation, we have to look to see if there is any way out starting from the middle of the masked window to the border, there are many possibilities and combinations of such path, in either meaning, we try to find a border that surrounding the centre background pixel, the pixel to be considered as a hole pixel if such border exists.

A: original text image, B: BIIA, C:SBIIA.

FIGURE 10: Another Example for Text Image.

Now, by applying our algorithm (SBIIA), Figure 11 shows the remedy of this problem by using of two different SE's as mentioned before.



A and B are the output of our algorithm on Figure 9A and Figure 9D respectively.

FIGURE 11: Application of SBIIA on Figure 9 (A and D).

As seen by latter figure, the number of hit points in Figure 11A is 4 points (including the reference point), and 6 points for Figure 11B, so, in both of these cases; the implementation of non-thresholding SE is quite enough, but, however, in other hand, the number of points in Figure 7C is 3 (including the reference point), for that reason we shift to implementation of thresholding SE for discovering and processing this situation and the decision will be applied.

8. PERFORMANCE EVALUATION

Table 1 shows the performance of our algorithm compared with BIIA, we have made a processing time comparison in order to reveal the speed of our algorithm as well as the accuracy as shown before.

Input	Dimension	BIIA (millisecond)	SBIIA (millisecond)	speed ratio %
Figure 8A	50x50	6	3	50
	200x200	484	142	29
	800x800	28528	8790	31
Figure 8D	50x50	13	4	30
	200x200	452	134	29
	800x800	28326	6764	23
Figure 10A	50x50	7	1	14
	200x200	343	89	26
	800x800	27811	5883	21
Blank Image	50x50	7	1	14
	200x200	430	14	3
	800x800	29381	346	1

TABLE 1: Time Factor Performance for BIIA and SBIIA.

As seen by Table 1, the speed up factor has a significant improvement and the overall algorithm has been improved as quality as well as the speed.

9. CONCLUSION

Hole filling algorithm plays a crucial role for object features extraction in order to be applied for recognition stage and the correct features comes from a perfect segmented object, so, segmentation operation may produce such kind of noise along with the existence of the embedded noise in the object, so, by holes filling algorithm we can overcome this problem and obtain a well segmented object, and other applications also for text document and other scanned objects.

We have modified BIIA and we have achieved both better quality and faster version of this algorithm, we have applied a dynamic SE filtering to improve the quantity of the output produced by original version of BIIA, the first SE applied was a general one and applied always which has zero values at the corners and one values elsewhere, the second one is all ones and applied in case of there are two hit points and one missed point out of 3x3 image sub-window which means there is a sharp corner point and special caring should be considered herein, the output of each step is the reference point will be set at the corresponding location in the image object after the above stipulation is been taken into consideration, this technique proved its reliability for filling the letters of the text images as well, the speed factor has been reduced to one third which allows speeding up the overall processing time in which this algorithm is a ring out of a processing chain.

10. REFERENCES

[1] R. Gonzalez, R. Woods. “Digital Image Processing”,Pearson Prentice Hall, First Impression, 2009, pp.665, 682.

[2] Oxford Dictionary.

[3] E. Firestone. “An Exploration Of Hole Filling Algorithms”, M.Sc. Thesis, Faculty Of California Polytechnic State University, San Luis Obisp, 2008.

[4] Z. Aktouf, G. Bertrand, and L. Perroton. (2002, March). “A Three Dimensional Holes Closing Algorithm”, ELSEVIER Pattern Recognition Letters, Vol. 23 (5), 523–531. Available: 10.1016/S0167-8655(01)00152-0.

[5] E. Cohen , L. Tekumalla , and E. Cohen. ”A Hole-Filling Algorithm for Triangular Meshes”, School of Computing, University of Utah, USA, 2004.

[6] M. Dellepiane, A. Venturi, and R. Scopigno. (2011). ”Image guided reconstruction of un-sampled data: a coherent filling for uncomplete Cultural Heritage models”, Springer Publication, International Journal of Computer Vision, USA, Vol. 94(1). Available: 10.1007/s11263-010-0382-2.

[7] M. Hasan, P. Mishra, “Connected Boundary RegionTracing for Object Hole Filling. ”International Conference on Advances in Engineering and Technology (ICAET 2011), Bhubaneswar, India, November 2011, pp.67-72.

[8] M. Droogenbroeck, and H. Talbot. (1996, December 30).”Fast Computationof Morphological Operations with Arbitrary Structuring Elements”, Pattern Recognition Letters Journal, USA, Vol. 17(14), 1451-1460. Available: 10.1016/S0167-8655(96)00113-4.

[9] X. Wu, M. Wang, and B. Han, “An Automatic Hole-Filling Algorithm for Polygon Meshes”, Computer-Aided Design and Applications, vol. 5(6):889-899, 2008.

[10] C. Chen, K. Cheng, and H. Liao, ”A Sharpness Dependent Approach to 3D Polygon Mesh Hole Filling”, Proceedings of EuroGraphics, 2005, pp. 13-16.

[11] A. Kumar, A. Shih, Y. Ito, D. Ross, and B. Soni,”A Hole-filling Algorithm Using Non-uniform Rational B-splines”, Department of Mechanical Engineering, University of Alabama at Birmingham, Birmingham, AL, U.S.A., >2008.

[12] K. Oh, S. Yea,and Y. Ho,” Hole-Filling Method Using Depth Based In-Painting For View Synthesis in Free Viewpoint Television (FTV) and 3D Video”, IEEE 27th conference on Picture Coding Symposium, Chicago, IL, 2009, pp. 1-4.

[13] A. Brunton, S. Wuhrer, C. Shu, P. Bose, and E. Demaine. (2010) “Filling Holes in Triangular Meshes Using Digital Images by Curve Unfolding”, International Journal of Shape Modeling, Vol. 16(1-2), 151-171. Available: 10.1142/S0218654310001328.

[14] M. Wei, J. Wu, and M. Pang. (2010). “An Integrated Approach To Filling Holes In Meshes”, IEEE International Conference on Artificial Intelligence and Computational Intelligence, 2010, Vol.3, 306 – 310. Available: 10.1109/AICI.2010.302.