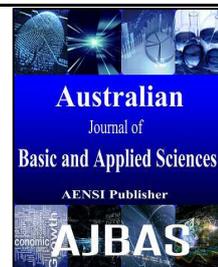




ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com



## Boundary Based Microarray Spot Segmentation using Iterative Incremental Foreground Expansion

Madhukar H. and Dr.Manjunath S.S.

Dayananda Sagar College of Engineering, Bangalore. India.

### ARTICLE INFO

#### Article history:

Received 10 November 2015

Accepted 30 December 2015

Available online 18 January 2016

#### Keywords:

8 surrounding pixels, microarray spot segmentation, morphological dilation, incremental foreground expansion.

### ABSTRACT

This paper presents a new method of microarray spot segmentation based on the spot boundary. Border pixels lying on the boundary of the foreground are inspected and processed for their suitability to merge with the present foreground. Those border pixels qualified to merge are included into the foreground for its incremental expansion. The boundary region of the present foreground is determined using the image dilation technique. Incremental foreground expansion is carried out iteratively until no further expansion is possible.

© 2015 AENSI Publisher All rights reserved.

**To Cite This Article:** Madhukar H. and Dr.Manjunath S.S., Boundary Based Microarray Spot Segmentation using Iterative Incremental Foreground Expansion. *Aust. J. Basic & Appl. Sci.*, 9(36): 360-364, 2015

## INTRODUCTION

Spot segmentation is an important step in cDNA microarray analysis (Demirkaya, O., 2005). It involves the separation of the Foreground and the Background pixels of the Microarray spot. Several methods are already available (Lehmussola, A., ; Dudoit, S., 2002; Rahnenfuhrer, J. and D. Bozinov, 2004). In a microarray spot image, the foreground occupies the central region having higher intensity pixels while the back ground occupies the remaining region having lower intensity pixels. Ideally, the foreground forms a circular connected region of higher intensity pixels. In practice, the shape is distorted, intensity levels are mixed up and noise/artefacts are introduced because of several unavoidable deficiencies in the process of generating the microarray images. Therefore the microarray segmentation is a complex task.

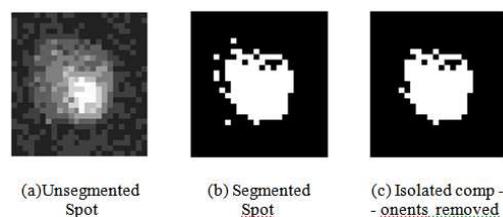
In our proposed method, segmentation is done in two stages. In the first stage, a simple and conservative spot segmentation algorithm is used to get the initial foreground region and background region. In the second stage, the present foreground is extended pixel by pixel wherever possible according to certain criteria which will be described later. It is essentially boundary based in the sense that those pixels on the outer boundary of the present foreground and lying on the background are the potential candidates for transfer from background to foreground.

### I. Basic Principle:

The microarray spot under consideration is taken as a gray scale image with pixel intensities varying from 0 to 255. The gray scale image can be obtained from the red channel, green channel or both depending on the user requirement.

#### A. Initial Segmentation:

Initial Segmentation is done by the binarization of the spot using Otsu threshold. Here, the threshold level OT is calculated using the Otsu method. Those pixels having intensities greater than OT are replaced by 1's (white) and all other pixels by 0's (black). Now, the white region represents the foreground and the black region represents the background. Fig.1(a) shows an example spot and Fig. 1(b) shows its initial segmentation in white and black. In Fig. 1(b) we see many small scattered bright spots isolated from the main foreground region. These isolated spots may not belong to the foreground.



**Fig. 1:** Results of initial Segmentation.

Because, even though their intensity levels are higher, their spatial disconnectedness disqualifies them for inclusion into the foreground.

For initial segmentation, other segmentation methods like k-means clustering, Gaussian Mixture Model, Seeded Region Growing etc. can be used.

### B. Removal of Isolated Pixels from the Foreground:

The small disconnected (isolated) components outside the main foreground region are removed using the connected component labelling. That connected component having the largest area is retained and all other smaller ones are discarded. After this operation, only the main foreground is left out. See Fig. 1(c). Let us call this as the initial foreground.

### C. Filling up of Interior holes:

If any interior holes are present in the initial foreground, they are filled up. Because, an interior hole, even though lower in intensity, is surrounded by the higher intensity foreground pixels and therefore qualifies to join the foreground region. Filling of the interior holes is done using morphological reconstruction techniques. Matlab function `imfill(BW, '8', 'holes')` is used to fill the interior holes. The result of filling the interior hole(s) for the example of Fig. 1 is shown in Fig. 2. In Fig. 2(a) an interior hole is present on the north by northwest corner which is filled up and the filled foreground is shown in Fig. 2(b). Other partial holes along the boundary which are not fully (connect-8) surrounded by the foreground are retained for further processing.

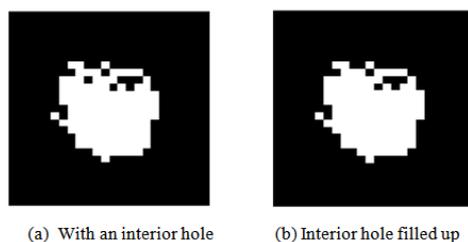


Fig. 2: Interior hole filling.

### D. Marking of the Outer Boundary of the Foreground:

The outer boundary of the initial foreground is determined as follows. We use morphological dilation (Gonzalez, R.C. and R.E. Woods, 2009) to extend the initial foreground all around by one pixel. Then the initial foreground is subtracted from the extended foreground to get the outer boundary as described below. Let matrix A represent the given gray scale plot (Kalannagari Viswanath and Ramalingam Gunasundari, 2015; Sindhu, T. and C. Nagavani, 2015). Let F be the binary image matrix corresponding to the initial foreground. The size F will be same as that of A. Let the structuring element

for image dilation be represented by the symbol 'se'. For one pixel thick dilation the structuring element needed is the square matrix of size 3x3 with all ones (Gonzalez, R.C. and R.E. Woods, 2009). It is represented in Matlab notation as, `se = strel('square', 3)`. In matrix notation se is given by,

$$se = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

Let the extended image matrix after dilation be G. Then G is given by,

$$G = \text{imdilate}(F, se) \quad (2)$$

Here, `imdilate(F, se)` is the image dilate function [9] which dilates F by the structuring element se. Then the outer boundary of F is given by

$$H = G - F \quad (3)$$

Here, H is the binary matrix which represents the pixels on the outer boundary of 1 pixel thickness. The sizes of G and H are same as that of A. The one pixel wide outer boundary corresponding to the foreground of Fig. 2(b) is shown in Fig. 3(b). The boundary is marked by white while the bare foreground in Fig. 3(a) is in gray.

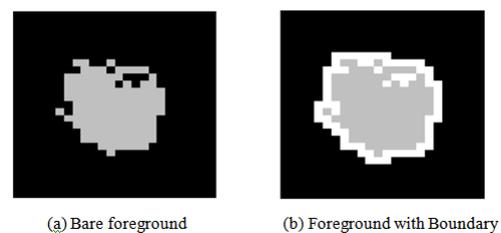


Fig. 3: Foreground with one pixel thick outer boundary.

### E. Border Pixels on the Boundary:

Those border pixels lying on the outer boundary of the foreground are further investigated to check whether some of them can join the foreground. Each border pixel is tested for certain eligibility condition. If a border pixel passes the eligibility condition, it is admitted into the foreground, else left in the background.

### E 8 Surrounding Pixels of Border Pixels:

Each border pixel is evaluated based on the status of its immediate 8 surrounding pixels. The immediate 8 surrounding pixels of four randomly selected border pixels are shown in Fig.4. The region between the outer and the inner square contains the 8-surrounding pixels of the corresponding border pixel. In our approach, the eligibility of a border pixel, to join the foreground, depends on the status of its 8-surrounding pixels (Raja Thilagam, A. and Dr. R. Suresh Babu., 2015).

### F. Classification of 8-surround pixels:

There are 8 surrounding pixels for a border pixel. Out of 8, some pixels lie on the foreground and the

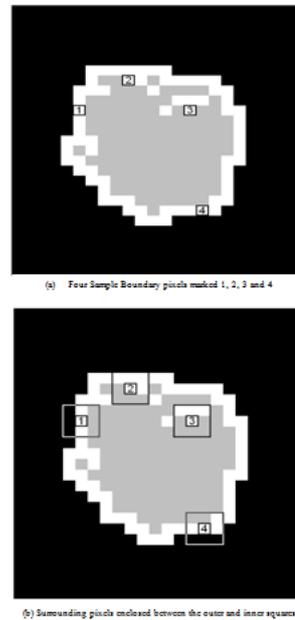
remaining on the background. Note that the boundary lies in the background because it is the outer boundary of the foreground. Thus for a border pixel, its surrounding pixels can be partitioned into two classes designated as *Fgclass* and *Bgclass*. Those lying on the foreground belong to the *Fgclass* and the remaining (those lying on the background) belong to the *Bgclass* (Dr. J. Subash Chandra Bose and Greeshma Gopinath, 2015).

**G. Class Count:**

The number of pixels in each class is called its *class count*. The number of pixels in the *Fgclass* is called the *Fgclass count* and similarly, the number of pixels in the *Bgclass* is called the *Bgclass count*. *Fgclass count* gives the number of foreground pixels adjacent to the corresponding border pixel while the *Bgclass count* gives the number of pixels adjacent to the background. In Fig. 4(b), we can see different cases for the class counts. For border pixel marked 1, see Fig. 4(a), 3 pixels ( east, northeast and southeast) belong to *Fgclass* and the remaining 5 pixels to the *Bgclass*. Thus, for this border pixel, *Fgclass count* = 3 and *Bgclass count* = 5. For the example of Fig. 4, the number of pixels (class count) in each class for all the four border pixels is shown in Table 1.

The eligibility of a border pixel to qualify for inclusion into the foreground depends on the status of its 8 surrounding pixels and the corresponding class counts. If a border pixel is eligible, it is labelled

as the foreground pixels and included into the present foreground. Then the next border pixel is examined for inclusion. This is the technique of incremental foreground expansion.



**Fig. 4:** Immediate 8-surrounding pixels for four border pixels.

**Table 1:** Number Of Pixels In *Fgclass* And *Bgclass* For Four Border Pixels Of Fig.4.

Serial No.	<i>Fgclass</i> Count. (Number of pixel in the <i>Fgclass</i> )	<i>Bgclass</i> Count (Number of pixel in the <i>Bgclass</i> )
1	3	5
2	4	4
3	5	3
4	2	6

**II. Incremental Foreground Expansion:**

In this approach, the relative value of the *Fgclass* count of a border pixel determines the eligibility of that border pixel to be labelled as a foreground pixel to join the foreground. *Fgclass* count can vary from 1 to 7. *Fgclass* count cannot be zero, because, in that case the corresponding border pixel has no adjacent (touching) foreground pixels. It is fully surrounded by background pixels only. This means the border pixel is not at all on the outer boundary of the foreground. If a pixel is to be a border pixel, it should touch at least one foreground pixel. Thus *Fgclass* count minimum = 1. *Fgclass* count cannot be 8 also. In this case the border pixel is fully enclosed by the foreground pixels. Hence, it cannot be a border pixel. Therefore, *Fgclass* count maximum =7. Thus the range of *Fgclass* count is from 1 to 7.

**A. Degree of Enclosure of a Border Pixel by the Foreground:**

Theoretically, *Fgclass* count of 8 means, the pixel under consideration is fully enclosed by the

foreground. The pixel is a hole. If the *Fgclass* count is 7, it means the pixel under consideration is almost enclosed. The degree of enclosure is 7 out of 8. We can thus define the *degree of enclosure* as the ratio of *Fgclass* count to 8, that is,

$$\text{Degree of Enclosure} = \frac{\text{Fgclass count}}{8} \tag{4}$$

**B. Principle of Incremental foreground Expansion:**

In a specific case, for a border pixel, let *Fgclass* count = 7. It means the corresponding degree of enclosure of that border pixel is 7/8. This means, it is 7/8 th surrounded by the foreground. Therefore, taking a liberal view, we are justified in labelling this as the foreground pixel and transferring this pixel to the foreground. This process is the incremental expansion of the foreground by 1 pixel. Similarly border pixels of *Fgclass* count of 6 with degree of enclosure 6/8, also can be classified into the foreground. Having assumed liberal bias towards foreground, we can stipulate that all border pixels whose degree of enclosure is greater than or equal to 0.5 are eligible to become the foreground pixels.

That is  $Fgclass\ count \geq 4$  makes the corresponding border pixel to be labelled as foreground pixels. Here the threshold count  $T$  is 4. That is, for a border pixel, If  $Fgclass\ count \geq T$ , label that pixel as foreground and include it into the foreground region. Thus the eligibility condition is,

$$Fgclass\ count \geq T \quad (5)$$

The incremental foreground expansion scheme can select other values for  $T$ , like 5, 6 or 7. Then the foreground expansion becomes more conservative, that is, lesser number of border pixels are included into the foreground. Incremental foreground expansion is carried out with respect to all the pixels on the present outer boundary. This we call as the first iteration.

### C. Multiple Iterations of Incremental Foreground Expansion:

The incremental foreground expansion, with respect to all the border pixels of the present boundary, forms one iteration. Once the foreground is expanded using the present boundary, it is ready for the next iteration. A new outer boundary is drawn around the present foreground and the incremental foreground expansion is carried out for all the border pixels on this boundary. This process is repeated iteratively until no more expansion is possible. In each iteration, one or more pixels are transferred from the background to the foreground. When Eq. (5) is not satisfied for any border pixel in an iteration, foreground expansion does not occur. No further expansion is possible and the iteration process is terminated.

### III. Iterative Incremental Foreground Expansion Algorithm:

The entire technique of the proposed spot segmentation is described in the following algorithm. Image dilation operation is used to get the outer boundary of the present foreground.

#### Algorithm 1:

INPUT: Microarray spot, a gray scale image.

OUTPUT : Segmented spot with clearly marked foreground and background.

1. Get the initial foreground region using Otsu threshold binarization.
2. Remove small isolated foreground regions by connected component analysis technique.
3. Fill the interior holes if any in the foreground region.
4. Get the one pixel thick outer boundary of this foreground.
5. Set the expansion count per iteration variable,  $ecpi = 0$ .
6. for each border pixel on this boundary,
  - a) Get the 8 surrounding pixels which are spread over both the foreground and background.

b) Examine and classify each surrounding pixel as either  $Fgclass$  (belongs to the foreground) or  $Bgclass$  (belongs to the background).

c) Determine the  $Fgclass$  count, the number of surrounding pixels in  $Fgclass$  and  $Bgclass$  count, the number of pixels in the  $Bgclass$ .

d) Set the threshold count  $T$  to 4.

e) If the  $Fgclass\ count \geq T$ , label the present border pixel as foreground pixel and include it into the present foreground. Now the foreground is incrementally expanded by one pixel. Increment the expansion count per iteration variable as,  $cpi = ecpi + 1$  Endfor. // one iteration over

7. If  $cpi = 0$ , it means no foreground expansion in the present iteration. Algorithm over. Exit. If  $cpi \geq 0$ , next iteration is needed. Go to step 4.

### IV. Experimental Result:

The spot shown in Fig. 4, is processed using the technique of Iterative Incremental Foreground Expansion. The resulting foreground, the outer boundary and the eligible pixels on the boundary are shown in Fig 5.

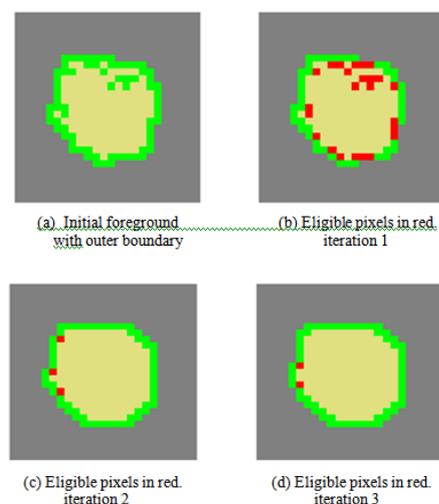
The foreground is shown in yellow, the boundary in green, background in dark gray and the eligible pixels are shown in red. Fig. 5(a) shows the initial foreground and the corresponding outer boundary. Fig. 5(b) shows the number of eligible pixels marked in red, which are merged with the foreground which can be seen in Fig. 5(c). The effect of incremental expansion in successive iterations are shown in Fig. 5(d), 5(e), 5(f) and 5(g). Fig. 5(h) gives the result when no more expansion is possible.

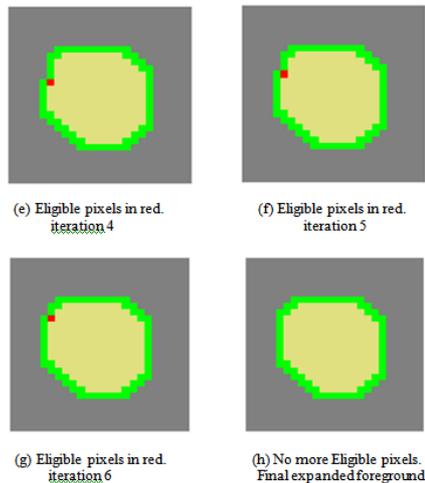
Total Number of pixels transferred to foreground = 32.

Total number of pixels in the image =  $27 * 24 = 648$ .

Total number of pixels in the final foreground = 152.

Percentage transfer =  $(32/152) * 100 = 21.05\%$ .





**Fig. 5:** Incremental segmentation by region filling.

### V. Conclusion:

A new technique of microarray spot segmentation is described. This method removes all the concave regions from the foreground and provides a smoothed boundary to it.

### REFERENCES

- Demirkaya, O., M.H. Asyali and M.M. Shoukri, 2005. "Segmentation of cDNA Microarray Spots Using Markov RandomField Modeling", *Bioinformatic*, 21: 2994–3000.
- Lehmussola, A., P. Ruusuvoori and Oi Yli-Harja, "Evaluating the performance of microarray segmentation algorithms", *Bioinformatic*, 22: 2910–2917.
- Dudoit, S., 2002. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sinica.*, 12: 111–139.
- Rahnenfuhrer, J. and D. Bozinov, 2004. Hybrid clustering for microarray image analysis combining intensity and shape features. *BMC Bioinformatics*, 5: 1–11.
- Otsu's method - Wikipedia, the free encyclopedia [en.wikipedia.org/wiki/Otsu's\\_method](http://en.wikipedia.org/wiki/Otsu's_method)
- Fill image regions and holes - MATLAB [www.mathworks.com/help/toolbox/images/ref/imfill.html](http://www.mathworks.com/help/toolbox/images/ref/imfill.html)
- Gonzalez, R.C. and R.E. Woods, 2009. "Digital Image Processing", third Edition. Pearson Education. Chapter, 9: 655-657.
- Gonzalez, R.C. and R.E. Woods, 2009. "Digital Image Processing", third Edition. Pearson Education. Chapter, 9: 651-652.
- [www.mathworks.com/help/toolbox/images/ref/strel.html](http://www.mathworks.com/help/toolbox/images/ref/strel.html)
- Viswanath, K. and R. Gunasundari, 2014. "Design and analysis performance of Kidney Stone Detection from Ultrasound Image by level Set

Segmentation and ANN classification". ISBN No: 978-1-4799-3080-7/14/2014 IEEE digital explorer, pp: 407-414. DOI: 10.1109/ICACCI.2014.6968485.

Kalannagari Viswanath and Ramalingam Gunasundari, 2015. "Analysis and Implementation of Kidney Stone Detection by Reaction Diffusion Level Set Segmentation Using Xilinx System Generator on FPGA," *VLSI Design*, Article ID 581961, 10. doi:10.1155/2015/581961.

Dr. J. Subash Chandra Bose and Greeshma Gopinath, 2015. A Survey Based on Image Encryption then Compression Techniques for Efficient Image Transmission. *J. Ind. Eng. Res.*, 1(1): 15-18.

Raja Thilagam, A. and Dr. R. Suresh Babu., 2015. Adaptive Time-Quantized pseudorandom Sampling scheme for Green Communication. *J. Ind. Eng. Res.*, 1(2): 22-28.

Sindhu, T. and C. Nagavani, 2015. Adaptive sequence of Key Pose Detection for Human Action Recognition. *J. Ind. Eng. Res.*, 1(3): 10-15.

Emimanothaya, A. and Dr. R. Suresh Babu., 2015. Maintaining Privacy and Integrity in two-tiered Wireless Sensor Network using Advanced SafeQ Protocol. *J. Ind. Eng. Res.*, 1(3): 16-23.