

Bacterial colony counting using distance transform

Dipti Prasad Mukherjee*, Amita Pal, S. Eswara Sarma, D. Dutta Majumder

National Centre for Knowledge Based Computing, Indian Statistical Institute, 203 Barrackpore Trunk Road,
Calcutta 700 035, India

Received 11 February 1994; accepted 22 August 1994

Abstract

A distance-transform based technique is presented for the segmentation of monochrome images of colonies grown on membrane filters. This is used to count the number of *Escherichia coli* in a given water sample, which is used as a parameter for determining water quality. The result is compared with fuzzy *c*-means clustering approach.

Keywords: Distance transform; Image processing; Pattern recognition; Fuzzy *c*-means clustering

1. Introduction

The quality of water used by communities is an important concern of users as well as administrators, since the general health of the community is greatly dependent on it. A certain class of bacteria, called coliforms, are reliable indicator organisms for testing water quality because they travel with disease-producing micro-organisms and are easy to isolate. Their presence in drinking water usually indicates that water is unsuited for drinking and may need treatment with a disinfectant. In a study [1], the World Health Organization (WHO) has proposed that the number of *Escherichia coli*, which is a coliform, could be a parameter for determining the quality of drinking water or coastal waters.

Most bacteriological water quality criteria are based on examining water samples for faecal pollution indicator bacteria, in particular, total and faecal coliforms. The membrane filter technique [2] is a relatively new and reliable laboratory technique for the bacteriological examination of water. Basically, the idea is to pass a known volume of the water sample under examination through a membrane filter, whose pore size is small enough ($0.45 \mu\text{m}$) not to allow *E. coli* and other coliforms to pass through. When this filter is incubated on an appropriate medium at a suitable temperature and for the right length of time (generally about 24 h), colonies develop upon the filter disk wherever bacteria were entrapped during the filtration process. Depending upon the medium used, the colonies of *E. coli* take on a colour which is different from those of other bacteria, so that they can be identified easily and counted, using either a low-power microscope or a magnifying glass. For the

present analysis, samples consisting only of organism type *E. coli* are considered. The monochrome images of 'blackish' coliform organisms in the 'whitish' background of the membrane filter are used for *E. coli* detection. Membrane filters may have grid lines, etched on them to facilitate the counting process.

The authors have made an attempt to develop a microcomputer-based system that can automate the counting, covering all the colonies on the filter in the process. Basically the steps involved are as follows:

1. Capture of the image of the membrane filter by means of a CCD camera.
2. Preprocessing of the image by the use of appropriate image processing techniques.
3. Development of algorithms for the classification and counting of the coliform colonies.

We have used distance transform technique [3] to count the number of *E. coli* components. This is detailed in the following section preceded by the image preprocessing technique. The algorithm is tested on several images some of which are reported in the result section. Its performance is compared with fuzzy classification approach reported in [4,5] followed by conclusion. The complete set up for the micro-computer based *E. coli* counting is shown in Fig. 1.

2. Problem definition

Broadly speaking, the problem is to identify and count bacterial colonies from images. This has

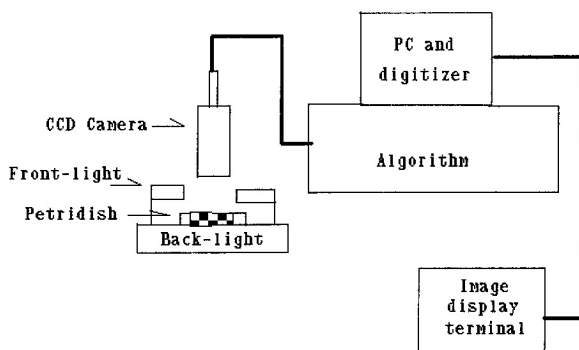


Fig. 1. Set-up for water quality analysis.

been reformulated by us as one of classifying pixels into one of three classes — colony, background and grid, the grid being, etched on the membrane filter by the manufacturers to aid the lab technicians in the counting process.

Complications arise on account of the presence of the grid on the membrane filter. While the grid serves as an aid for quick counting of colonies for the human operator, in our case, that is, when using Image Processing/Pattern Recognition (IP/PR) techniques, it poses problems, if we treat it (the grid) as a separate class, in addition to background or colonies. At present, the grid cannot be eliminated, since we require it to prove and calibrate the performance of our algorithm, as compared to the manual counting. This is discussed in subsequent sections.

3. Preprocessing

At least eight different views of the same image of each petridish are taken under varying front-lighting conditions. This is done to reduce the effect of specularly from colonies and non-uniform lighting over the viewing surface. Images for all the plates are captured under a constant magnification. Though adhoc, we have found the following steps to work satisfactorily:

(i) The average pixel values of the eight views are calculated.

(ii) The sum of the absolute differences of the average pixel value from the pixel values of the individual slides scaled linearly from 0 to 255 scale gives a satisfactory enhanced image for colony identification. The reason for taking the difference of the average from the individual slide is to increase the contrast between colony class and the other class consisting of the grid and the background. It is expected that the average pixel values of the grid and the background for all the views will be approximately close to the pixel values of the grid and the background of individual slides, whereas the pixel values of coliforms for different views will be different particularly due to specularly effect. Reflection of light from colony is non-uniform because of its irregular locally non-planar spherical surface (which is not the case either for the grid or for the background).

Therefore, the difference will try to decrease the pixel value of the grid and the background class in contrast to colony class. The absolute difference is taken so that the pixel values after summing up are non-negative (since pixel value of an image could not be negative). Mathematically speaking, given, pixel value p_{ijk} , for the i th slide at position jk of the 2D image matrix, and the average pixel value of the eight different views of the petridish, at jk th position, \bar{p}_{jk} , $\sum_{i=1}^8 |p_{ijk} - \bar{p}_{jk}|$, scaled as mentioned earlier, gives the enhanced pixel value.

However, we are in the process of designing a proper set-up for capturing the images of colonies with both back- and front-lighting, so that non-uniformity of lighting over the entire area is somewhat reduced. We intend to follow [6]. We hope that this step can be eliminated once we have our specialized set-up for capturing images of colonies.

4. Using distance transform

The steps involved in making colony counts, based on the distance transform technique, are as follows:

1. Thresholding of the preprocessed image.
2. Application of the distance transform to the image.
3. Thresholding of the distance-transform space, the local maxima obtained in 3×3 neighborhood being grown to get back the almost equivalent size of the colony or its congruence. The grid lines are eliminated due to thresholding in the distance-transform space.
4. Application of the connected component labelling algorithm to count the number of colonies.

The gray level thresholding is straightforward and it is done interactively based on user's choice; the only consideration being that there is no loss of colony components in the thresholded image. Average gray value of a portion of filter paper image without grid line and coliform organisms determines the threshold limit. This is because we want to convert the image matrix to two classes — one representing colony and grid class and the other being the background. The grids are subsequently eliminated from the colony and the grid class using distance transform. An automatic

threshold selection method is also implemented based on Otsu's method [7]. The optimum threshold value is evaluated based on inter and intra-class variances when thresholding divides the entire gray levels of the image into two classes: information (the colony and the grid class) and background. The goodness of the threshold at level k is determined based on the following discriminant criterion:

$$\lambda = \sigma_B^2 / \sigma_W^2 \quad \kappa = \sigma_T^2 / \sigma_W^2 \quad \text{and} \quad \eta = \sigma_B^2 / \sigma_T^2 \quad (1)$$

where σ_W^2 , σ_B^2 and σ_T^2 are the intra-class variance, inter-class variance and the total variance of gray levels respectively. Interested readers may refer to [7] for detailed calculation. The problem of searching the optimum threshold is reduced to an optimisation problem to find best k which maximises these discriminant functions. In fact, it can be shown that η is the simplest and appropriate measure to evaluate the goodness or separability of the threshold at level k .

It is noted that automatic threshold selection may not be always reliable but the threshold value obtained following automatic technique could be used as a good starting point for interactive thresholding.

After thresholding, the binary image has pixel values 0 and 1, 1 representing colony or grid, while 0 is the background. The distance transform of the thresholded image can be achieved in two passes [3] only. For a 3×3 mask

$$\begin{array}{ccc} A & B & C \\ D & E & F \\ G & H & I \end{array}$$

in the first pass, starting from the top-left corner of the segmented image, the central pixel E is replaced by

$$E_{\text{new}} \leftarrow \min(D, A, B, C) + 1 \quad (2)$$

In the second pass, originating from the bottom-right corner, the central pixel E is replaced by

$$E_{\text{new}} \leftarrow \min(G, H, I, F, (E - 1)) + 1 \quad (3)$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	1	1	2	1	1	0	0
0	0	1	1	2	2	2	1	0	0
0	0	1	2	2	3	2	1	0	0
0	0	1	2	3	3	2	1	0	0
0	0	1	2	3	3	2	1	0	0
0	0	0	1	2	3	2	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	1	1	2	1	1	0	0
0	0	1	1	2	2	2	1	0	0
0	0	1	2	2	3	2	1	0	0
0	0	1	2	2	2	2	1	0	0
0	0	1	1	2	2	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

Fig. 2. (a) An example image of size 9×10 . The pixel value 1 denotes information while 0 is the background. (b) Result of applying first pass of distance transform on the image of Fig. 2(a) following equation (2). (c) Distance-transform space for Fig. 2(a) after completing the second pass following equation (3) on the image of Fig. 2(b).

Note that both the above conditions are applied only when the central pixel value E is 1.

An example is shown in Fig. 2. Fig. 2(a) is an image with nine rows and ten columns. The pixel value 1 is the information whereas 0 is the background. After completing the first pass following Eq. (2), the output is shown in Fig. 2(b). Fig. 2(c) is the final output after distance transform (distance-transform space) applying the second pass following Eq. (3).

The local maxima in the distance-transform space gives the central or core point of each colony or its congruence. The local maxima is calculated in 3×3 neighborhood. For example, the local maxima with distance transform value 3 is the core point at (5,6) of Fig. 2(c) if the distance-transform space of image of Fig. 2(a) is thresholded at 2. Since, the thickness of the grid line is constant throughout the image space, the distance-transform value of the grid line is constant. This gives the value for thresholding the distance-transform space eliminating grid lines.

The core points are then expanded to a $(2k - 1) \times (2k - 1)$ square, where k is the value of the core point which remains in the centre of the square. Note that while expanding, the grown region may contain other core points. A linked list data structure is implemented to keep tag on the connected components.

For a set $\{S\}$ containing a core point p_i , taken from set $\{U\}$ which contains all the core points, iterate the following steps:

A. For core point $p_i \in \{S\}$ with distance-transform value k , select the square $(2k - 1) \times (2k - 1)$, with the core point p_i being at the centre of the square. By selection, we mean setting the value of all the pixels within the square $(2k - 1) \times (2k - 1)$ to 1 and search for any other core point within the square $(2k - 1) \times (2k - 1)$.

B. If there are core points p_{i1}, \dots, p_{im} in the square $(2k - 1) \times (2k - 1)$, add them to set $\{S\}$. Subtract all these core points from set $\{U\}$.

C. Eliminate p_i from set $\{S\}$.

D. If $\{S\} \neq 0$ go to step A and continue iteration till $\{S\} = 0$.

E. If $\{S\} = 0$, all the selected pixels due to iterations of the above steps represent a single connected component. The size of the connected

component is given by the number of pixels with value 1. In this case ($\{S\} = 0$), add another $p_i \in \{U\}$, not yet considered, to set $\{S\}$ and continue iteration from step A for the next connected component until all the core points (p_i) of $\{U\}$ are considered.

Representing the above steps in psuedo-code, let $value(p_i)$ gives the value of the core point p_i in the distance-transform space $dspace[1...N][1...N]$. Suppose, set $\{U\}$ contains n core points of the distance-transform space, i.e., the local maximas found in $dspace[1...N][1...N]$ after thresholding. $\forall p_i \in \{U\}, i \in \{1, n\}$ and $value(p_i) > 0$, initialize a set $\{S\} = \{p_i\}, i = 1, \{S\} \subset \{U\}$

Begin loop on i :

$k \leftarrow value(p_i)$

Begin loop for $u = -(k - 1)$ to $(k - 1)$

Begin loop for $v = -(k - 1)$ to $(k - 1)$

if $dspace[u][v]$ is a core point p_{uv}

add p_{uv} to $\{S\}$ and subtract p_{uv} from $\{U\}$

$dspace[u][v] - 1$

End loop for u

End loop for v

Subtract p_i from $\{S\}$

If $\{S\} \neq 0$ then continue loop on i

If $\{S\} = 0$ then

increase number of connected component by 1; size of connected component = number of pixels with value 1; add another $p_i \in \{U\}$ to $\{S\}$ and begin loop on i for next connected component until $\{U\} = 0$.

Due to this region-growing (selection of square $(2k - 1) \times (2k - 1)$ for central core point with distance-transform value k), approximate colony structure can be recovered though the exact topology will be somewhat different due to digital analysis of the colony shape.

The segmented image is then subjected to connected component labelling algorithm [8], which is as follows:

1. All the dark pixels (corresponding to colonies) are marked with integers (starting from 1) in increasing order. The image is scanned in the increasing order of rows, starting from its top-left corner.

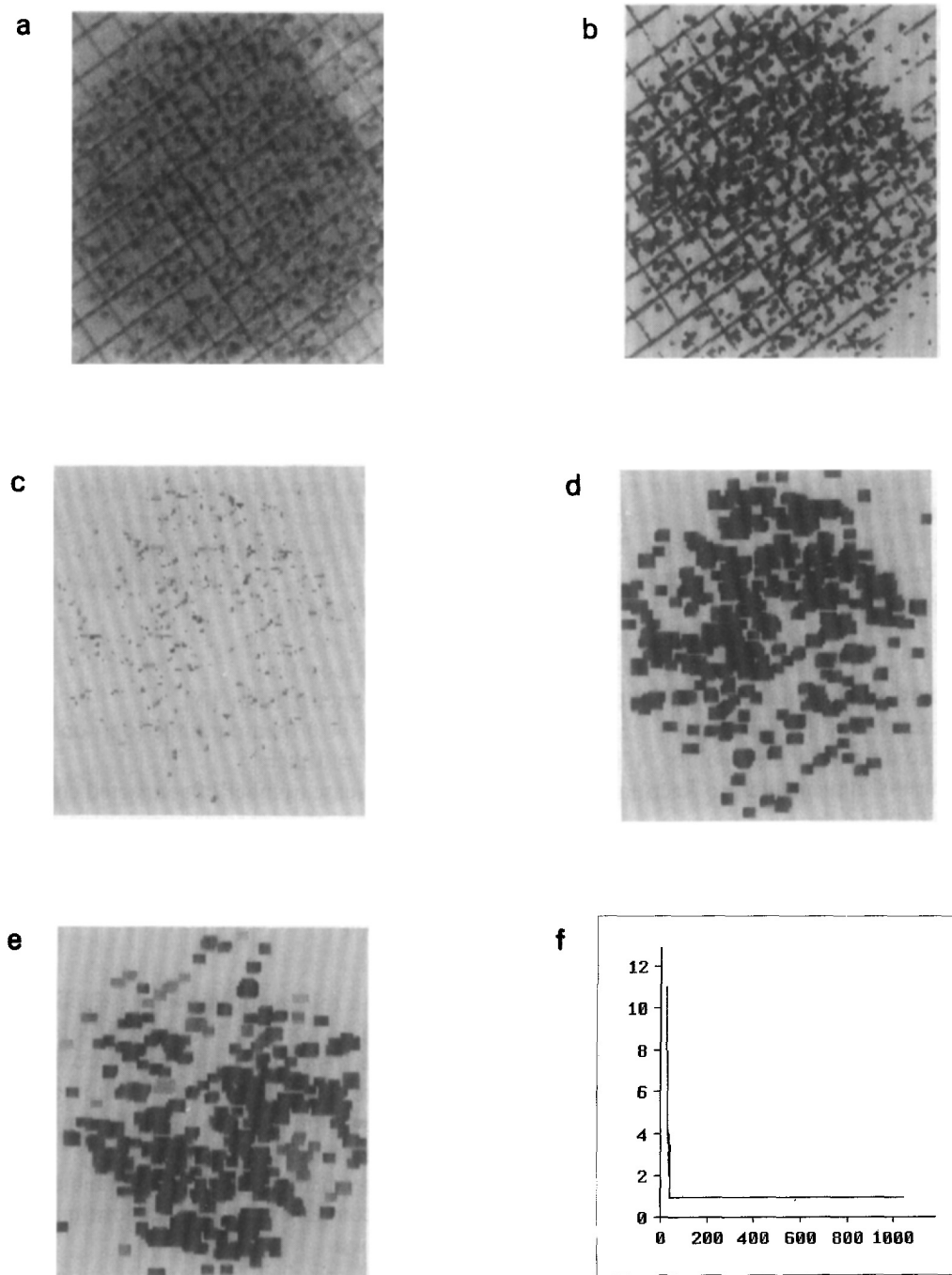


Fig. 3. (a) Original image. (b) Image in (a) after thresholding at gray value 125. (c) Image in (b) after distance transform. (d) Image in (c) after region growing. (e) Image in (d) after component labelling. (f) Frequency distribution of size of components of the image in (e). Mode of distribution is approximately at 43 and the number is 398.

2. In the subsequent iterations, each pixel and its marked value is replaced with the local maxima in an 8-connected neighborhood. The iterations continue till there is no further replacement for all the dark pixels.

3. Count the occurrences of each pixel marking values from its frequency distribution.

4. The frequency distribution of the count in step 3 gives the number of colonies in the given image. The mode of the distribution gives the size of a single colony after the specified duration of incubation, since it appears logical to assume that single colonies appear in far greater numbers than overlapping colonies. The number of colonies present is calculated by dividing the total number of dark pixels by the mode of the distribution.

5. Results

We test the distance transform algorithm on image of Fig. 3(a). Fig. 3(b) gives the image after thresholding, while Fig. 3(c) shows the local maxima after thresholding the distance-transform space, eliminating the core points for grid lines. Fig. 3(d) shows the effect of region growing as described above, maintaining the connected component topology. Fig. 3(e) shows the output after labelling of all the connected components, while

Fig. 3(f) gives the distribution of colony size. It is observed that, even with congruences of colonies, the number of occurrences of single colony (without congruence) in the image, is much higher compared to the same of its (colony) congruences. Therefore, from the mode of the size distribution, as shown in Fig. 3(f), the size of a single colony of Fig. 3(d) is 43. The total number of black pixels, representing colony class, divided by the size (in number of pixels) of a single colony gives the colony count which is 398 for Fig. 3(a). Several such counts are carried out and the algorithm using distance transform seems to give results close to $\pm 10\%$ of the operator counted value. The coliform count by an expert operator for all the images is given in Table 1.

Also, Figs. 4(a) and 5(a) are segmented images with front- and back-lighting, whose original images are Figs. 6(a) and 7(a), respectively. Note that there is no grid present in case of back-lighting (Fig. 7(a)). Figs. 4(b) and 5(b) give the local maxima of the distance-transform space duly thresholded after eliminating core points of the grid lines, particularly for front-lit images. Figs. 4(c) and 5(c) give the result after connected component labelling. The counting results are described in Table 1. Readers can compare the results of Figs. 4(c) and 5(c) with Figs. 6(b) and 7(b) which are obtained using fuzzy *c*-means classification.

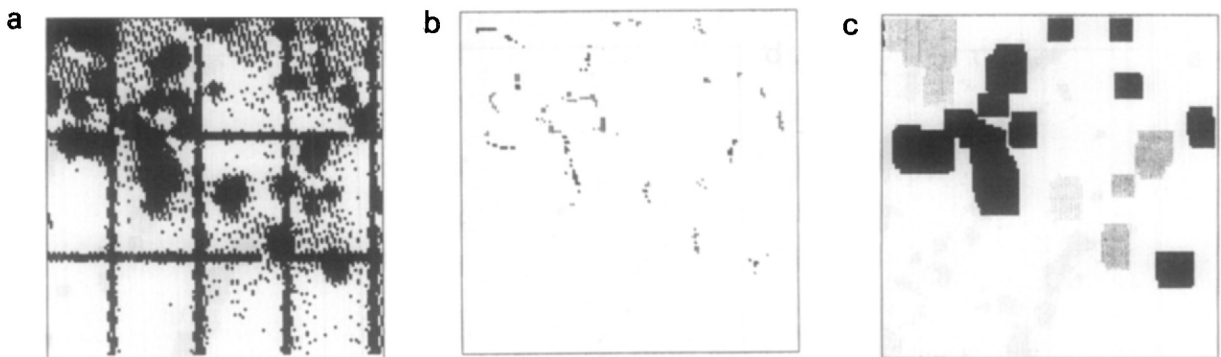


Fig. 4. (a) The image in Fig. 6(a) after thresholding at gray value 115. (b) The image in Fig. 4(a) after finding local maxima of thresholded distance transformed space. (c) The image in Fig. 4(b) after region-growing and component labelling.

Table 1
Overall result

Fig. No.	Manual count	Coliform count	Details
3(a)	420	398	using distance transform
4(a)	47	42	using distance transform
5(a)	44	37	using distance transform (B)
6(a)	47	72	using fuzzy <i>c</i> -means clustering $c = 3, v = 2, m = 2$
7(a)	44	42	using fuzzy <i>c</i> -means clustering (B) $c = 3, v = 2, m = 2$

c = class, v = feature vector, m = fuzzy exponent, B = backlit.

6. Comparing distance transform with fuzzy *c*-means clustering

In [4], we have formulated the colony counting as a classification problem. The three distinct classes present in the slides are the colonies, the grid and the background. The features associated with each pixel are

1. the average of gray values in a 3×3 mask, and

2. the busyness value which is a parameter indicating the direction of variation of gray values — horizontal or vertical. Since, the grid lines present in the filter paper have dominant directional characteristics, we expect that busyness value will show this directional bias.

The busyness [9] of each pixel is defined as $\min(v_x, v_y)$ where

$$v_x = |A - B| + |B - C| + |D - E| + |E - F| + |G - H| + |H - I| \quad (4)$$

$$v_y = |A - D| + |D - G| + |B - E| + |E - H| + |C - F| + |F - I| \quad (5)$$

and

$$\begin{matrix} A & B & C \\ D & E & F \\ G & H & I \end{matrix}$$

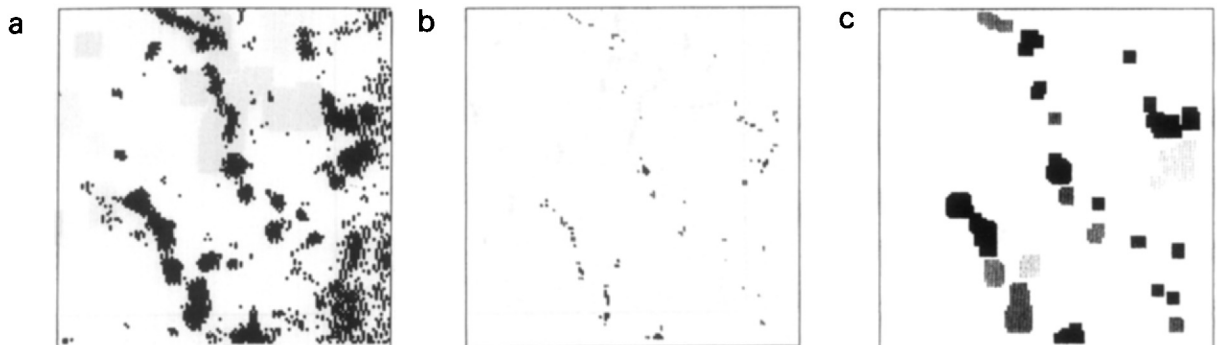


Fig. 5. (a) The image in Fig. 7(a) after thresholding at gray value 200. (b) The image in Fig. 5(a) after finding local maxima of threshold distance transformed space. (c) The image in Fig. 5(b) after region-growing and component labelling.

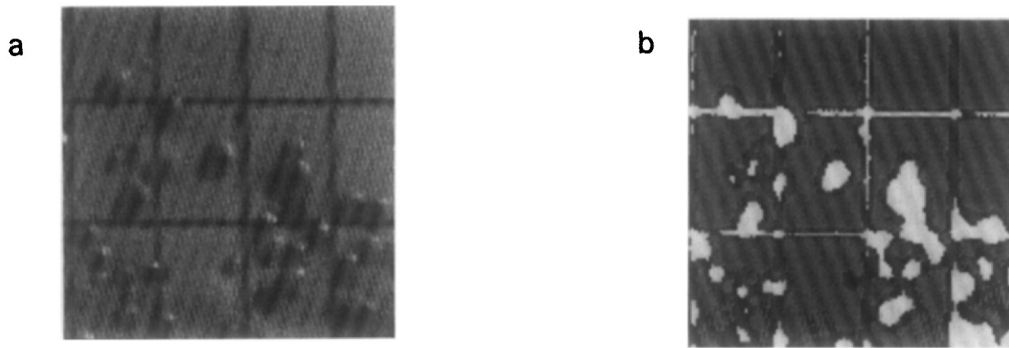


Fig. 6. (a) Original image. (b) The image in (a) after fuzzy *c*-means clustering with three classes and two vectors.

is the 3×3 mask where the busyness value is calculated for the pixel E . Instead of $\min(v_x, v_y)$, we found that the busyness value equivalent to $\max(v_x, v_y)$ for the three classes mentioned above has more discriminating power.

The standard implementation of fuzzy *c*-means algorithm [9] is followed to discriminate the bacterial colonies from the grid and the background.

Figs. 6(a) and 7(a) are the output of the preprocessing step, the fuzzy *c*-means classification of which are shown in Figs. 6(b) and 7(b), respectively. Note that there are three classes and two feature vectors, as mentioned earlier. We have taken fuzzy weighting exponent value to be 1.5.

Based on the accuracy, repeatability and the processing time, the distance transform method is more stable compared to the fuzzy *c*-means approach. The extent of misclassification of grid class, partially to other classes, depends on the training set and gives a wide variation in the result. In both the cases, the user interaction is minimum. Automatic selection of the threshold value of distance-transform space, based on the values of core points of the grid lines, is quite straightforward as the width of the grid line is constant for all the images and known apriori. For fuzzy *c*-means clustering, we found that the use of seed-points of a set of images to an almost similar set

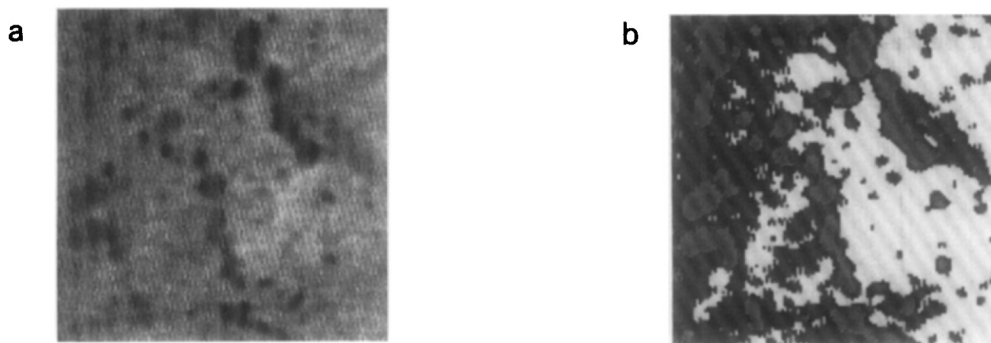


Fig. 7. (a) Backlit original image. (b) The image in (a) after fuzzy *c*-means clustering.

is not always advisable and may give unsatisfactory results, thereby demanding learning even for a minor variation in the images.

7. Conclusions

We hope to achieve better performance by tuning several steps of the algorithms mentioned above. Most of the samples tested are within the acceptable limit of accuracy in comparison to the results obtained by an experienced human operator. For example, for Fig. 3(a) where the coliform count using distance transform is 398, manual count is approximately 420. The detail comparative study on this will be reported in a subsequent publication (Mukherjee Dipti Prasad et al. in preparation). Some of the issues on which we are concentrating now are as follows:

1. An improved and faster connected component labelling algorithm.
2. Modified region growing algorithm. Note that our objective here is to differentiate between a single colony grown to a larger size from a congruence of colonies.
3. Tuning up of fuzzy *c*-means algorithm with a large number of samples. This process will be an attractive alternative if stabilized ready-to-use seed points for the colonies, the background and the grid lines could be provided. An additional feature vector, with linear structural information, may give better discrimination in classification.

Acknowledgments

We would like to thank Anjana Dewanji, Nikhil Ranjan Pal, Bhabatosh Chanda and Punam Saha

for their invaluable assistance in the course of this work. We would also like to thank D.P. Pal and U. Chatterjee of the All-India Institute of Hygiene and Public Health for providing samples and domain knowledge. Finally, we thank the anonymous referees for their critical review and thoughtful suggestions. This work was partially supported by DoE/UNDP (IND/85/072).

References

- [1] Health criteria and epidemiological studies related to coastal water pollution. Report of a group of experts jointly convened by WHO and UNEP, WHO Regional Office for Europe, Copenhagen, 1977.
- [2] Pelczar MJ, Reid RD and Chan ECS: *Microbiology*, Tata McGraw-Hill, New Delhi, 4th edition, 1983.
- [3] Arcelli C and deBaja S: A width-independent fast thinning algorithm. *IEEE Trans Pattern Analysis Machine Intelligence*, PAMI-7 (1985) 463–474.
- [4] Pal Amita, Mukherjee Dipti Prasad, Sarma SE and Majumder D Dutta: Fuzzy set theoretic image processing techniques in the bacteriological analysis of water samples. In *Proceedings of 1st Asian Fuzzy Symposium*, Singapore, 1993.
- [5] Mukherjee Dipti Prasad, Pal Amita, Sarma SE and Majumder D Dutta: Water quality analysis: a pattern recognition approach, *Pattern Recognition*, (1994) in press.
- [6] Chan MK, Kwok KC, Shuen YM and Koo KL: Development of a microcomputer-based system for the automatic recognition of *E. coli* colonies, *Lab Microcomput*, (1990) 95–101.
- [7] Otsu Nobuyuki: A threshold selection method for gray level histogram, *IEEE SMC*, SMC-9 (1979) 62–66.
- [8] Rosenfeld A and Kak AC: *Digital Picture Processing*, Academic Press, New York, 1982.
- [9] Bezdek JC: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.