

# Image Gridding Algorithm for DNA Microarray Analyser

Maziidah Mukhtar Ahmad  
School of Microelectronic Eng.  
University Malaysia Perlis  
Perlis, Malaysia  
jidan8189@gmail.com

Asral Bahari Jambek  
School of Microelectronic Eng.  
University Malaysia Perlis  
Perlis, Malaysia  
asral@unimap.edu.my

Mohd Yusoff bin Mashor  
School of Mechatronic Engineering  
University Malaysia Perlis  
Perlis, Malaysia  
yusoff@unimap.edu.my

**Abstract:** A deoxyribonucleic acid (DNA) microarray is a powerful tool that is widely used in genetics to monitor the expression levels of thousands of genes in parallel. Gridding, segmentation and intensity extraction is the process of gene expression. In the process of gene expression, gridding process information leading to dig their own DNA and provide coordinates for each point. Gridding can be implemented by using three kinds of process which is manual, semi-automatic and fully automatic. Well-organized and easy automatic gridding method for DNA microarray image analysis was proposed in this paper. Five popular techniques for DNA microarray gridding are evaluated with highlighted on gridding accuracy, processing time and capability of noise removal.

**Keywords:** *image processing; microarray; gridding; algorithm; accuracy*

## I. INTRODUCTION

In 1995 microarray technology was created by M. Schena et al. [1] and from that it has been used as a significant tool in the study of genes expression. The single chip with unique locations for each spot contains abundant levels of DNA. This allows for the simultaneous estimation of the expression of thousands of genes. DNA can be used to identify hidden biological information [2], classified the treatment for drug addict [3] and nowadays famous uses in cancers genes identification. By making differential and analysing for both normal and abnormal microarray gene expression profiling, disease can be identified.

Three main steps for microarray processing are used which is: gridding, segmentation and intensity extraction. Errors can be detected at an early stage in the process of gridding. So gridding is a very important process in this experiment to avoid inaccurate results [4]. Microarray image segmentation is a difficult process and takes a long time to solve a problem because the process involves images contain noise, the image with low intensity, object is not homogeneous and poor contrast [5]. This also affects feature extraction accuracy. In microarray image processing, gridding is essential for assigning coordinates for each spot. Spots will be separated from each other by horizontal and vertical grid lines. Three microarray gridding methods provided by looking at the amount of human intervention needed namely: manual, semi-auto and

automatic. Although each method tenders a number of dissimilar gridding accuracy, these methods share the same matters as misalignment of the grid and a high level of image noise. This will affect the image gridding accuracy and thus to extract precise DNA information.

This paper highlights some of the microarray gridding methods have been suggested in the literature. The rest of the paper is structured as follows: section two reviews the existing algorithm method for gridding microarray; section three discusses the methodology as it concerns the algorithm; section four highlight the results pertaining to algorithm; section five perform the comparison between algorithm and final section serves as a conclusion.

## II. LITERATURE REVIEW

An efficient and fully automated method for gridding microarray images [6] can classify all spots devoid of any human intervention, based on data driven that one-time human setup. For this method, the author applied preprocessing for the gridding of all spots. Additional steps were needed prior to getting the correct grid for each spot. The proposed preprocessing steps concerned applying histogram equalization in the form of a Wiener filter to obtain a high contrast between foreground and background noise and to compute the mean horizontal profile. All steps were conducted using an autocorrelation method. Figure 1 shows the differences achieved with and without preprocessing. The advantage of this method is that it can remove image noise.

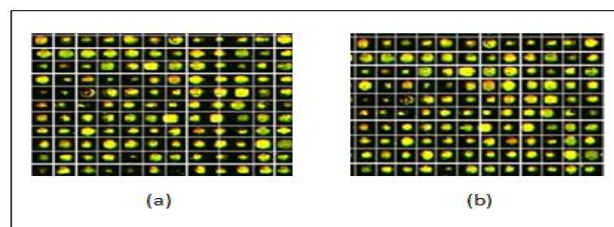


Figure 1: Differences between (a) with preprocessing and (b) without preprocessing [6].

DNA microarray gridding that uses conditional convolution sub-block histogram equalization [7] is implemented in two steps: converting the image into a binary image using the histogram transformation function and then applying a conditional convolution filter. This method is able to localize each spot using sub-block histogram equalization, as shown in Figure 2 and then filter the image to remove unwanted area. In this case, Blekas' automatic microarray gridding method is adopted. The method works by summarizing the red and green intensities along the horizontal and vertical lines to create global profiles of the spots. These profiles result in a signal that has multiple peaks and each peak corresponds to the coordinates of a row or column of spots. Global vertical and horizontal profiles are determined by calculating the midpoint of two successive peaks between rows and columns. A local histogram is applied to stretch image contrast. This local histogram equalization is converted to a binary image to separate the wanted signal and unwanted signal. Equation (1) is used to perform histogram equalization by choose a mean value of pixels in every sub-block.

$$T(r_k) = \sum_{j+Mi}^k p_i(r_j) = \sum_{j+Mi}^k \frac{n_j^i}{n^i} \quad (1)$$

Where

$T(r_k)$  represents the transformation function

$i$  represents the sub-block

$n_j^i$  represents the number of pixels in the sub-block

$n^i$  represents the total number of pixels in the sub-block

$n_j^i / n^i$  represents the probability density function of the sub-block

$M_i$  represents the mean of the pixels in the sub-block

The authors of Developing a New Methodology for De-noising and Gridding cDNA Microarray Images [8] conducted an experiment using images in which various conditions of noise occurred. One gridding process which starting by applying the autocorrelation of mean horizontal profile and by calculating the maximum peak indexes are the best gridding achievement. Then, the estimated period for receiving the distance between two adjacent spot centres was calculated. Gridding implementation process would become easier and the proper grid if the steps above are followed.

Automatic gridding of DNA microarray images using optimum subimage [9] include two major steps in the form of preprocessing and parameter calculation. For preprocessing, the author set one image as a binary reference image of the original image. As usual, the preprocessing step begins with adaptive histogram equalization; following on, spot regions are distinguished by applying canny edge detection; the final step is to fill the holes. Once all steps have been completed, the mean intensity value of pixels is calculated at each position. Optimum sub-images are considered for block with the maximum mean intensity, as shown in Figure 3.

Best gridding process that more accurate and fast was suggested compared to the mathematical morphological method. First, the image quality was enhanced using logarithm transformation. Then, an optimal threshold was

gained based on the Otsu method. The Otsu method is implemented by converting a colour image to a grayscale image and sharpening the image using edge processing. The converting process is done by separating the background and foreground of the image. This way, it is easier to remove unwanted parts of the image. The Otsu method [10] served as the best method for gridding each spot separately, without any noise or unwanted signals. The Otsu process is described in detail in Figure 4.

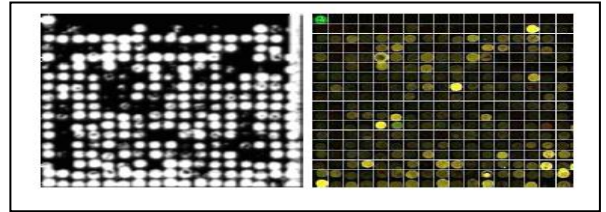


Figure 2: Conditional convolution sub-block[7]

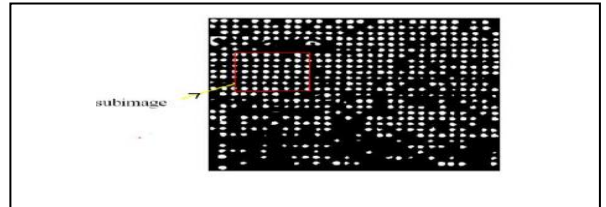


Figure 3: Optimum sub- image[9].

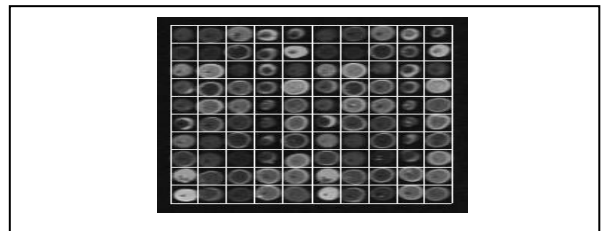


Figure 4: Otsu image [10].

Table 1 presents a comparison of each method to highlight the differences between them.

Table 1: Comparison of algorithm steps for the gridding method.

| METHOD/ FEATURES                 | PROCESS             | [6]                | [7]                       | [8]                     | [9]                    | [10]                |
|----------------------------------|---------------------|--------------------|---------------------------|-------------------------|------------------------|---------------------|
| Read Image                       | Pre-processing      | /                  | /                         | /                       | /                      | /                   |
| Crop Image                       |                     | /                  |                           |                         |                        | /                   |
| Separate Plane                   |                     |                    |                           |                         |                        | /                   |
| Convert RGB to Grayscale image   |                     | /                  |                           | /                       | /                      | /                   |
| Convert RGB to Binary image      |                     |                    | /                         | /                       | /                      |                     |
| Estimate Spot by Autocorrelation |                     | /                  |                           | /                       |                        | /                   |
| Locate Centers                   |                     | /                  | /                         | /                       |                        | /                   |
| Histogram Equalization           | Spot identification | (Wiener Filtering) | (Conditional Convolution) | (Morphological Opening) | (Canny Edge Detection) | (Top Hat Filtering) |
| Draw Gridline                    | Gridding            | /                  | /                         | /                       | /                      | /                   |
| Techniques                       |                     | Automatic          | Automatic                 | Automatic               | Automatic              | Automatic           |
| Noise removal                    |                     | Yes                | Yes                       | Yes                     | Yes                    | Yes                 |

### III. METHODOLOGY

Figure 5 shows the process for completing the gridding process. This process was divided into three parts: pre-processing, starting from reading the image up to converting RGB to greyscale; the second part was spot identification using autocorrelation, with the final part being gridding. Pre-processing is extremely important, as it can assist in removing unwanted signals or noise. Doing so can also simplify the second part of the process, i.e., spot identification. This part starts with the read image. Matlab can only read certain image formats. For this experiment, the JPEG image format was used. Once the image has been loaded and read, the next step is to crop the image. Cropping the image is important for simplifying the experiment, as it focuses on the image in a block-by-block manner.

We used the crop function in Matlab to crop a certain part of the picture that needed to be identified. To crop a specified region, we first needed to confirm region coordinates, so that we could use the command 'imcrop (x [coordinate])', as shown in Figure 6.

Figure 7 shows how the plane was separated into three planes (red, green and yellow); the split image was then saved in RGB format. For this part, we needed to extract the DNA sample image into three layers, because we wanted to identify the three layers' intensity (red spot intensity, green spot intensity and yellow spot intensity). Once this was achieved, the three layers' intensities could be compared for the following step. Custom colour maps render visualization more intuitive. Notice that spot shapes are not necessarily the same in each colour. The following Matlab command was used to separate planes:

```
redMap = gray(256);
redMap(:,[2 3]) = 0;
subimage(x(:,1),redMap)
axis off
title('red (layer 1)')
```

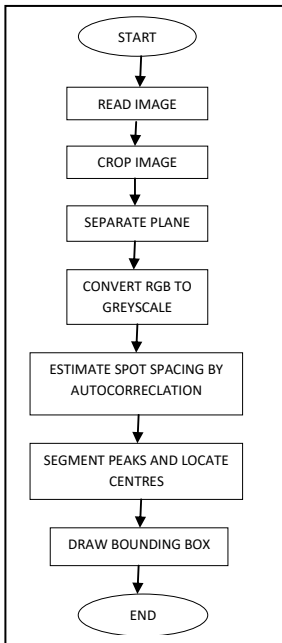


Figure 5: Flow chart.

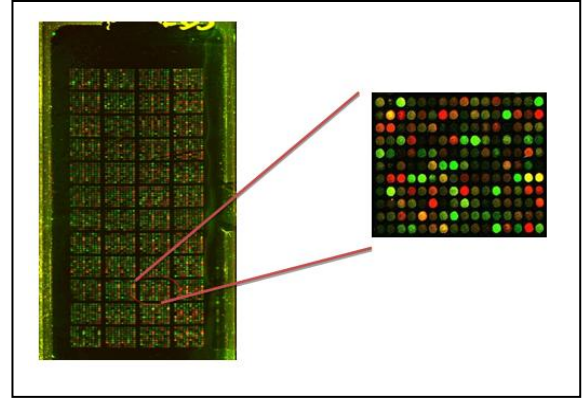


Figure 6: Original and cropped image.

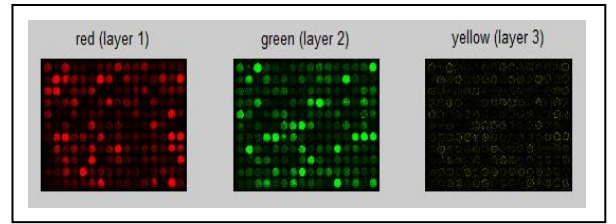


Figure 7: Split into three layers (red, green and yellow) from the cropped image.

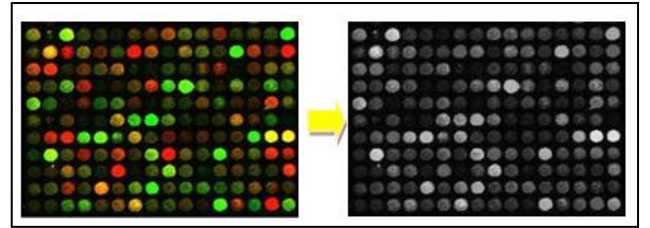


Figure 8: Colour image (RGB) to greyscale image.

From the cropped image, the RGB image was converted to a grayscale image for spot finding and for easily drawing a gridline (Figure 8). A regular grid of spots is also very important in this part. At this point, we were able to observe the intensity for each column of the image. To convert the RGB image to grayscale for spot finding, we used 'z=rgb2gray(y)'. Ideally, we assumed all the spots there should be periodically located in consistently; however, all spots were different in terms of intensity and size. The projection profile method was then applied to the binary image. Horizontal and vertical profiles were used to get rid of noise and receive the distance of spot [11]. For simplifying the calculation, the horizontal and vertical projections are defined as in (2):

$$\begin{aligned}
 Hp(y) &= \frac{1}{X} \sum_{x=0}^{X-1} f(x, y) \\
 Vp(x) &= \frac{1}{Y} \sum_{y=0}^{Y-1} f(x, y)
 \end{aligned} \tag{2}$$



Where  $HP$  is the horizontal profile,  $VP$  is the vertical of the  $f(x,y)$  image.  $X, Y$  stands for the amount of the image spot independently. The sum of intensity values are calculated at each pixel along the x-axis for each row, which is defined as follows:

$$MH(y) = \frac{1}{X} * \sum_{x=0}^{X-1} f(x,y) \quad (3)$$

where  $MH(y)$  mean horizontal profile of the image,  $f(x, y)$  dimensions  $X$  and  $Y$ , pixel  $x, (x, y)$ . Negative peaks profile that an appropriate to the position of the vertical grid lines are detected. If existing gridding process continues to be used, noise and other factors that exist in the real image can cause the phenomenon of loss or excessive grid lines. Therefore, de-noising and smoothing the profile projection is needed to make the grid correctly. From the peaks values of the autocorrelation profile like Figure 9, the mean horizontal profile was enhancement using top-hat filtering. Top-hat filtering only performs on grayscale or binary image input. Filtering was function as noise or artefact removal. The divergence among the image and opened version known as top-hat filter. It improves the details that would otherwise are hidden in low contrast areas.

When all noise from background had been removed, the next step is segments peaks or as a vertical and horizontal separators. Figure 10 shows the vertical separators and Figure 11 indicates the located centre. From the value of the mean profile projection, the gap between two spots could be identified. Gridline or bounding box could be drawn correctly as shown in Figure 12 by finding the mid- point between the two spots from right and left, and between the bottom and top spots.

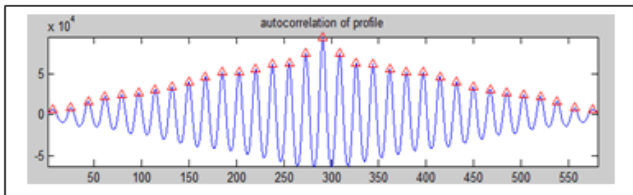


Figure 9: Autocorrelation Profile

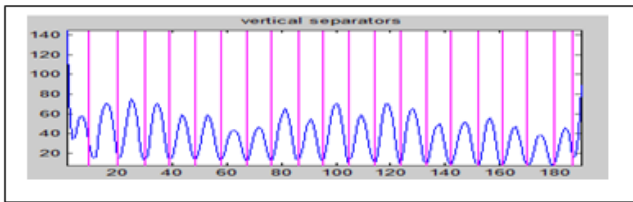


Figure 10: Vertical separators.

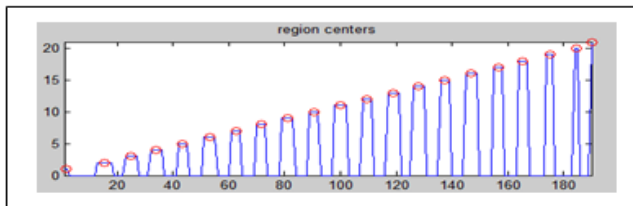


Figure 11: Region centres.

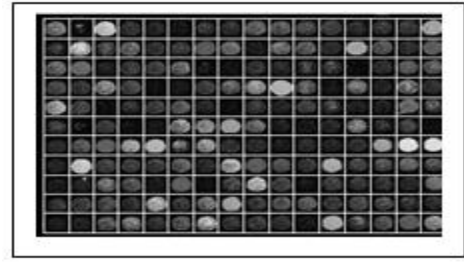


Figure 12: Bounding box.

#### IV. RESULTS AND DISCUSSION

An image from the Stanford Microarray Database (SMD) that has 176 spots was cropped. Following the image crop, the preprocessing process was begun. The image was converted to greyscale to find the spot more easily. Figure 13 below shows the process of the Otsu gridding method.

Horizontal profiling is a one step process for looking at a regular grid of spots. With computing the mean intensity for every line of the image, the centres can be identified and the space among gaps can be measured, as shown in Figure 14.

Autocorrelation in Figure 15 was used to enhance the self-similarity of the profile. The smooth result promoted peak finding and the estimation of spot spacing.

The spacing estimates were used to help design a filter for removing background noise from the intensity profile. The Top-hat filters were used to remove noise from the image, as shown in Figure 16. In Figure 15 and Figure 16, we can see the different signal after noise removal. Figure 16 shows this clearer than Figure 15.



Figure 13: Colour image to greyscale image.

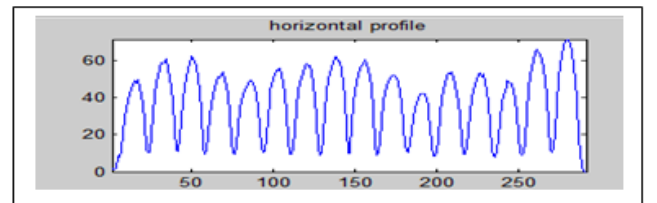


Figure 14: Mean horizontal profile.

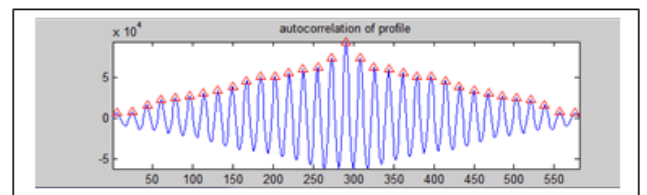


Figure 15: Autocorrelation of profile.

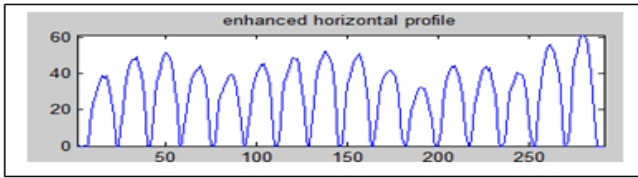


Figure 16: Enhanced horizontal profile.

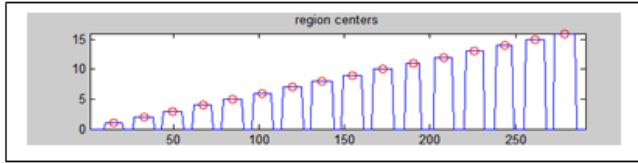


Figure 17: Region centres.

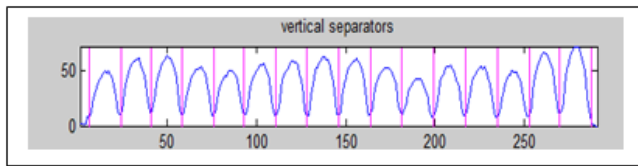


Figure 18: Vertical separators.

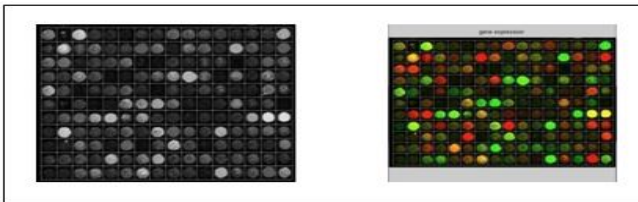


Figure 19: Results of gridding.

The next step, the gap between the top of the image and bottom, is important. We can extract the centroids of the peaks; these correspond to the horizontal centres of the spots. By using "regionprops" function in Matlab, common blob analysis can be performed for feature extraction. Once completed, we can determine the divisions between spots by providing grid point locations in the midpoints between adjacent peaks (see Figure 17 and Figure 18).

Rectangular grids were found following the completion of the preprocessing step. Using pairs of neighbouring grid points, we were able to form bounding box regions to address each spot individually. To facilitate the experiment, the size and position coordinates of each bounding box scheduled into four columns ROI (regions of interest) matrix (see Figure 20).

```

ans =
     6     5    18    18
     6    23    18    17
     6    40    18    18
     6    58    18    18
     6    76    18    17
     6    93    18    17
     6   110    18    18
     6   128    18    18
     6   146    18    17
     6   163    18    17

```

Figure 20: Four-column ROI matrix.

The accuracy of image gridding was calculated using (4), where respectively  $N_{\text{total}}$  spots and  $N_{\text{correct}}$  spots stand for the total number of spots in images and number of spots correctly gridded. Processing time was defined as the time taken for Matlab to complete the computation. These programs were simulated using a laptop computer (processor: Intel(R) Core (TM) i3- 2.10GHz, 2.00 GB RAM).

$$A = (N_{\text{correct spots}} / N_{\text{total spots}}) * 100\% \quad (4)$$

## V. PERFORMANCE COMPARISON

This experiment was conducted using various images. Images were classified into three types: best image, typical image and worst image. All images were described by deterministic grid geometry, known background intensity with zero uncertainty, pre-defined spot shape (morphology) and constant spot intensity that (a) was different from the background; (b) was directly proportional to the biological phenomenon (up-or down-regulation) and (c) had zero uncertainty for all spots [12]. Table 3 and Table 4 show the full results of this experiment, which was implementing for three types of images (Figure 21, Figure 22 and Figure 23). All of the results in the table were taken from our own simulations.

From Table 3 and Table 4, we can summarize that method [10] yielded the best accuracy and computational time compared to the other five methods for all types of images. The second best method was method [6], while the worst was method [7]. The top-hat algorithm can work best in greyscale images, where it can remove all unwanted signals, thus resulting in accurate gridding. However, in some images, a Wiener filtering algorithm can create a more accurate grid when compared to the top-hat method. This is because when the image is converted to greyscale, some spots will be missed or unclear. Since Wiener filtering identifies the edges of spots, the potential for missing spots is low. A conditional convolution gives the lowest accuracy compared to the other five algorithms. Method [10] has lower computational time compared to the other methods.

## VI. CONCLUSION

In this paper, existing microarray gridding methods were analysed and compared. For the algorithm part, the five best methods were discussed and the advantages and disadvantages of each method were presented. Based on our analysis, methods that utilized noise removal prior to performing image gridding yielded higher accuracy (>90%) and provided the best images, compared to methods that did not perform noise removal. Among the reviewed methods, the Otsu method yielded the best results, with 98% reported accuracy within 1 second. Our experiment was conducted using various microarray images in order to confirm this result. In the next phase, we plan to implement portable biomedical application with more effectively. To achieve this target, a robust analysis of the algorithm and architecture image DNA analyser should be carried out first.

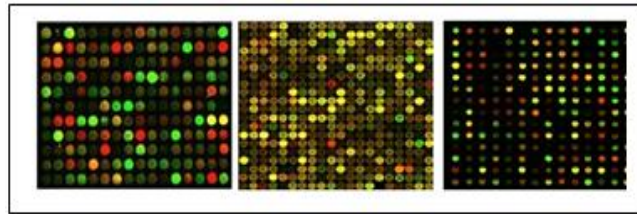


Figure 21: Best image types.

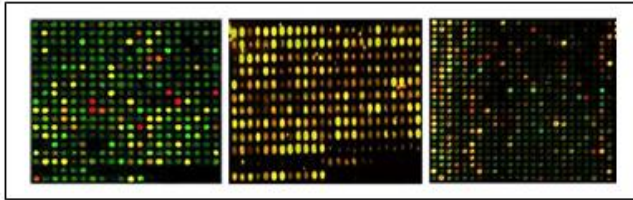


Figure 22: Typical image types.

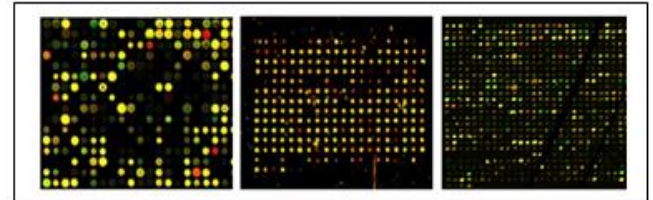


Figure 23: Worst image types.

Table 3: Accuracy results.

| METHOD | ACCURACY (%) |    |    |         |    |    |       |    |    | AVERAGE |
|--------|--------------|----|----|---------|----|----|-------|----|----|---------|
|        | BEST         |    |    | TYPICAL |    |    | WORST |    |    |         |
| [6]    | 64           | 75 | 78 | 28      | 23 | 13 | 17    | 32 | 35 | 40.56   |
| [7]    | 14           | 26 | 35 | 20      | 21 | 11 | 13    | 11 | 10 | 17.89   |
| [8]    | 36           | 30 | 41 | 24      | 28 | 17 | 24    | 21 | 20 | 26.78   |
| [9]    | 68           | 54 | 66 | 32      | 30 | 25 | 18    | 16 | 11 | 35.56   |
| [10]   | 98           | 90 | 96 | 32      | 77 | 17 | 29    | 71 | 89 | 66.56   |

Table 4: Time processing results.

| METHOD | PROCESSING TIME (SEC) |      |      |         |      |      |       |      |       | AVERAGE |
|--------|-----------------------|------|------|---------|------|------|-------|------|-------|---------|
|        | BEST                  |      |      | TYPICAL |      |      | WORST |      |       |         |
| [6]    | 4.91                  | 2.33 | 2.52 | 4.65    | 4.34 | 5.29 | 5.58  | 5.05 | 6.34  | 4.56    |
| [7]    | 7.63                  | 8.55 | 7.07 | 8.99    | 7.34 | 7.52 | 13.06 | 9.16 | 12.66 | 9.11    |
| [8]    | 3.05                  | 3.95 | 2.91 | 4.00    | 3.98 | 4.10 | 4.04  | 4.44 | 4.36  | 3.87    |
| [9]    | 3.24                  | 4.43 | 4.16 | 3.78    | 4.50 | 6.03 | 4.57  | 5.10 | 7.82  | 4.85    |
| [10]   | 1.04                  | 1.02 | 0.98 | 2.31    | 2.72 | 2.25 | 3.00  | 2.04 | 2.12  | 1.94    |

## REFERENCES

- [1] Schena M., Shalon D., Davis R.W., and Brown P.O. 'Quantitative monitoring of gene expression patterns with a complementary DNAmicroarray'. *Science*, 270, 1995, pp. 467-470.
- [2] Guillermo, H. L., R. L. Alajandro, A. S. Jose, P.V. Beatriz and FernandoM. S. 'Microarray and colon cancer in the road for translationalmedicine'. *Current Bioinformatics*, June 2011 .
- [3] Debouck, C., and Goodfellow, P.N. 'DNA Microarray in DrugDiscovery and Development'. *Nat Genetics*, 1999 .
- [4] White, A. M., Daly D. S., Willse A.R., Protic M., and Chandler D.P.'Automated microarray image analysis toolbox for MATLAB'. *OxfordUniversity Press, Oxford*, 2005.
- [5] Solomon, C., and Breckon T. *Fundamentals of digital image processing:a practical approach with examples in MATLAB*. 2011 .
- [6] F. El-Zahraa Labib, I. Fouad, M. Mabrouk, and A. Sharawy, "An Efficient Fully Automated Method for Gridding Microarray Images," *Am. J. Biomed. Eng.*, vol. 2, no. 3, pp. 115–119, 2012.
- [7] L.Pap,J.Jia Zou,"Image Enhancement for DNA Microarray Gridding using Conditional Convolution Sub-Block Histogram Equalization," pp. 618–623, 2007.
- [8] I. A. Fouad, "Developing a New Methodology for De-noising and Gridding cDNA Microarray Images," pp. 142–145, 2012.
- [9] T. Thomas and Q. Array, "Automatic Gridding of DNA Microarray Images using Optimum Subimage 1, 2," *Int. J.*, vol. 1, no. 4, pp. 37–40, 2009.
- [10] H. Tian, S. K. Lam, and T. Srikanthan, "Using Area-Time Efficient Logarithmic Approximation Unit," *Architecture*, pp. 0–3,2003.
- [11] I. Fouad, M. Mabrouk, and A. Sharawy, "Automatic and Accurate Segmentation of Gridded cDNA Microarray Images Using Different Methods," vol. 4, no. 2, pp. 41–54, 2014.
- [12] P. Bajcsy, L. Liu, and M. Band, "DNA Microarray Image Processing," *DNA Array Image Anal. Nuts Bolts (Nuts Bolts Ser.*, pp. 1–77, 2007.