Journal of Computational Intelligence in Bioinformatics
ISSN: 0973-385X Volume 3 Number 2 (2010) pp. 185–198

http://www.ripublication.com/jcib.htm

Automatic Technique for Gridding of Skewed and Noisy Microarray Images

J. Nagaraja[1], S.S. Manjunath[2] and Lalitha Rangarajan[3]

[1]Lecturer, [2]Assistant Professor,
[1,2]Department of Computer Science,
Dayananda Sagar College of Engineering, Bangalore, India
E-mail: [1]nagaraj.dsce@gmail.com, [2]mnj_ss2002@ yahoo.co.in
[3]Reader, Dept. of Studies in Computer Sci., University of Mysore, India
E-mail: lali85arun@yahoo.co.in

Abstract

Gridding is the one of the major processing stage in microarray image analysis. The existing approaches for gridding of microarray images require human intervention, which reduces the performance of whole gridding system. In this paper, fully automatic rectangular grid approach for gridding of skewed and noisy microarray images is presented. The proposed approach estimates the spatial coordinates required to construct rectangle grid around each spot in a subgrid. Also at the initial stage uses skew estimation algorithm based on corner spots detection to detect and correct any skew in the subgrid and adaptive threshold, which is used in filtering process to eliminate artifacts and noise. Experiments on Stanford and TBDB illustrates robustness of the proposed approach in the presence of skew, artifacts, noise and weakly expressed spots.

Keywords: Microarray, Gridding, Adaptive threshold, Spatial topology, Rectangular grid, Noisy images.

Introduction

DNA microarray technology has a large impact in many application areas, such as diagnosis of human diseases and treatments (determination of risk factors, monitoring disease stage and treatment progress, etc.), agricultural development (plant biotechnology), and quantification of genetically modified organisms, drug discovery, and design. In cDNA microarrays, a set of genetic DNA probes (from several hundreds to some thousands) are *spotted* on a slide. Two populations of mRNA,

tagged with fluorescent dyes, are then hybridized with the slide spots, and finally the slide is read with a scanner. The outlined process produces two images, one for each mRNA population, each of which varies in intensity according to the level of hybridization represented as the quantity of fluorescent dye contained in each spot.

Microarray image processing consists of the following sequence of three main tasks 1. Gridding, separation of spots by assignment of image coordinates to the spots. 2. Segmentation, separation between the foreground and background pixels and 3. Intensity extraction, computation of the average foreground and background intensities for each spot of the array.

Gridding is an important task that is to be performed as accurately as possible, since it affects the subsequent tasks of segmentation, intensity extraction and finally the conclusions derived out of the whole analysis. The available gridding software packages Scanlyze [1], Dappel [2], Image Gene[3], Genepix and SpotFinder[4] require human intervention in order to specify input parameters as well as to adjust properly the location of the grid lines. The template based approach is most prevalent in the existing packages which require specification of parameters such as spot size, spot spacing and space location. Some software products already incorporate an automatic refinement search for grid location, given size and spacing of spots [2,3]. Irregular grids cannot be found with the template based approach unless the template is manually adjusted to fit predefined distortions [3]. Automating this part of the process is essential because it reduces error in grid that may arise due to inaccurate specification.

The problem of automatic gridding is complicated because microarray images are usually highly contaminated with the noise and artifacts of the wet lab processes. There is a high need for automated methods for microarray gridding which are robust and flexible at the same time.

Some efforts on automatic microarray gridding have been reported in literature. However most of them impose different kinds of restrictions and are based on stringent assumptions. For example, the approaches in Jain et al.[5] and Yan et al.[6] requires that the grid rows and columns are strictly parallel to the x and y image axes. Other approaches, such as described by Carstensen et al.[7] and Katzer [8], rely on the Bayesian paradigm to deal with uncertainty and noise. Some well-known approaches to gridding microarray images are based on axis projections (Deng et al.[9]), or on morphological filtering (Yan et al.[10]). Both of them require user intervention in order to manually adjust the grid location. The Hill-Climbing approach for automatic gridding (Rueda et al.[11]) can perform gridding properly only if misalignments and rotations of the ideal grid are not present. Markov random field (MRF) (Antoniol et al.[12]) and graph-based grid approaches (Jung et al.[13]) have been used to perform gridding. A drawback of these approaches is that they require input parameters. A variety of different methodologies have been proposed with the intension to solve rotation and misalignment problems. Bajcsy [14] has suggested an exhaustive search of all the expected rotation angles, where as Steinfath [15] has estimated the rotation angle. A drawback of these approaches is that, it introduces pixel distortions when the rotation angle is small. Brandle et al.[16] utilized the discrete Radon transformation to estimate the angle of rotation. As it is computationally expensive, the process is

accelerated by constraining the range of rotation angles. Ho et al.[17] expressed the gridding process as an optimization problem based on the Jacobi iteration. However, this method is efficient only when the grids are smoothly distorted. Giuiliano et al.[18] recommended a gridding procedure based on stochastic search algorithms. Although it deals with rotations effectively, it requires manual intervention in order to define the radius of the spots.

In spite of the potential importance of gridding in microarray image analysis, the existing gridding approaches pay little attention on pre-processing of noisy microarray images and focus mainly on spot localization and spot segmentation. But the proposed approach is noise-resistant and it is effective, even under adverse conditions such as varying spot sizes, artifacts and weekly expressed spots.

Figure1 shows a block diagram which describes the salient stages of the proposed approach for automatic gridding of microarray images.

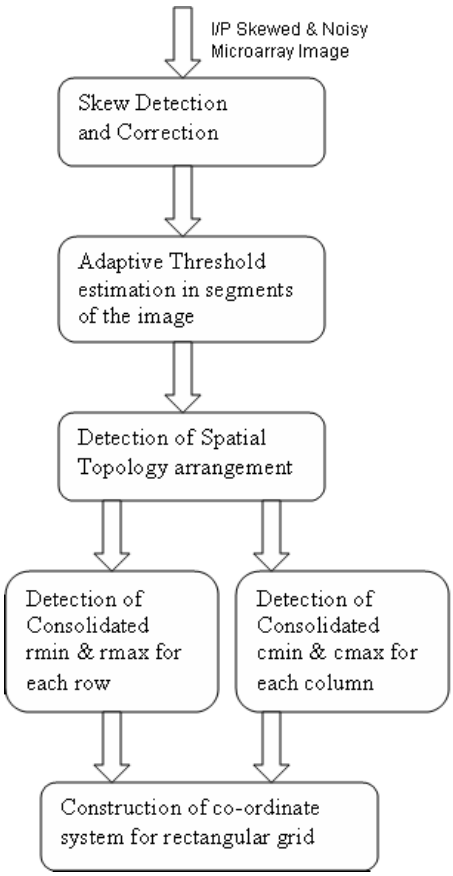

Figure 1: Stages of automatic gridding of noisy microarray image.

The organization of rest of the paper is as follows: Section 2 presents skew detection and correction algorithm, section 3 describes adaptive threshold technique. In section 4 automatic gridding processes which consists of detection of rmin & rmax, detection of cmin & cmax and gridding method are described. Section 5 highlights the

results of extensive experimentation conducted on some benchmark images. Finally conclusion is discussed.

Skew detection and Correction

In this section algorithm to detect skew in subgrid image based on center point (spot corner) method is presented.

***Steps to detect skew angle*:**

1. Converting input subgrid image to grey scale image.
2. Detecting each spot from 4 corners according to grey level color intensities.
3. Finding centroids of the each spot.
4. Connecting centriods of corner spots.
5. Calculating the skew degree for every iteration.
6. Saving results and moving to adjacent spots and repeating steps until end of spots (Midpoint of subgrid).
7. Linear regression analysis is performed to find actual skew angle.

The four spot corners are connected to each other with their centroids as shown in figure 2. Horizontal spots are connected by horizontals lines (A and B && C and D) and vertical spots are connected by vertical line (A and C && Band D) as shown in figure 2.

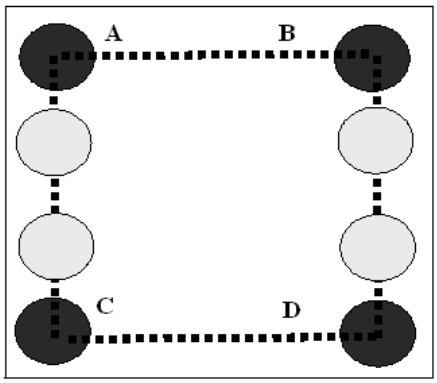

Figure 2: Centriods connected spots.

The following equation is used to find skew angle α based on figure 3. T is a relative point of the origin of image. It appears when skew is detected, position of T depends on, which spot is more deviated with respect to other spot.

$$dis\tan ce = \sqrt{(Xj - Xi)^2 + (Yj - Yi)^2} \quad (2.1)$$

$$c = \sqrt{(Yi - Yj)^2} \quad (2.2)$$

$$\alpha = \arcsin(c / dis\tan ce) \quad (2.3)$$

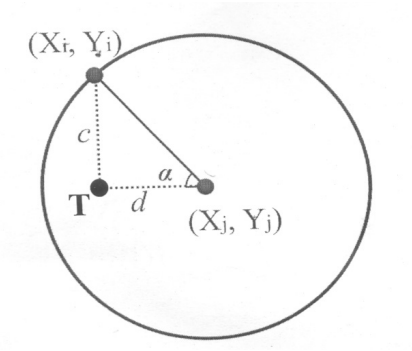

Figure 3: Trigonometry of spots interconnection.

According to figure 3 , (Xi,Yi) is current position of a first spot (Xj,Yj) is current position of second spot , α is degree of deviation

The above equation is used to detect skew angle on first four corner spots in the first iteration. In the next iteration, the same operation is performed on the next four corner spots ,the direction of movement is shown in figure 4 and whole process is continued until the midpoint of subgrid is found. The proposed algorithm is efficient, even in the absence of spots at any corner while performing the skew detection. If any of the spot is missing at any corner out of four spots (Horizontal/Vertical corner spot), the iteration moves to next subsequent rows / columns and direction of movement is shown in figure 4.

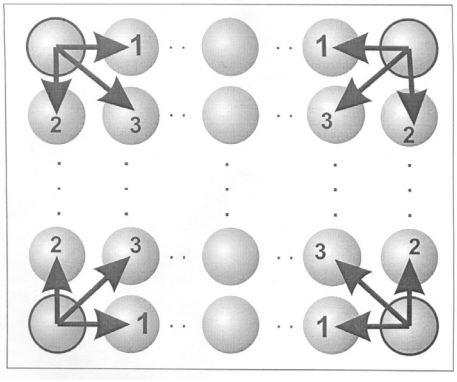

Figure 4: Pointer Movements.

Linear regression analysis is performed to derive most accurate results. Horizontal and vertical analysis is made to achieve correct results as shown in figures 5 and 6. Figures 7 and 8 (Image ID: 75757) shows the clockwise skewed image and skew corrected image.

Figures 9 and 10 (Image ID: 39114) shows the anticlockwise skewed image and skew corrected image. Table 1 illustrates the skew degrees found for different images (Data Bases) using proposed approach.

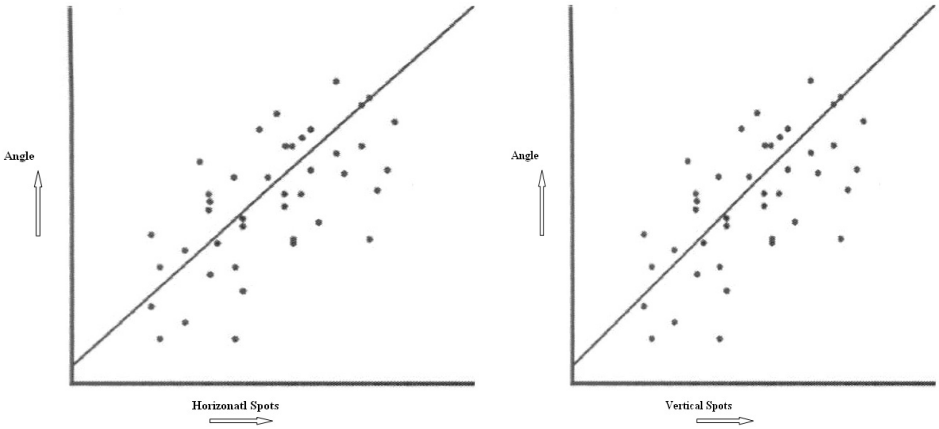

Figure 5 and 6: Linear regression curve for Horizontal and Vertical spots.

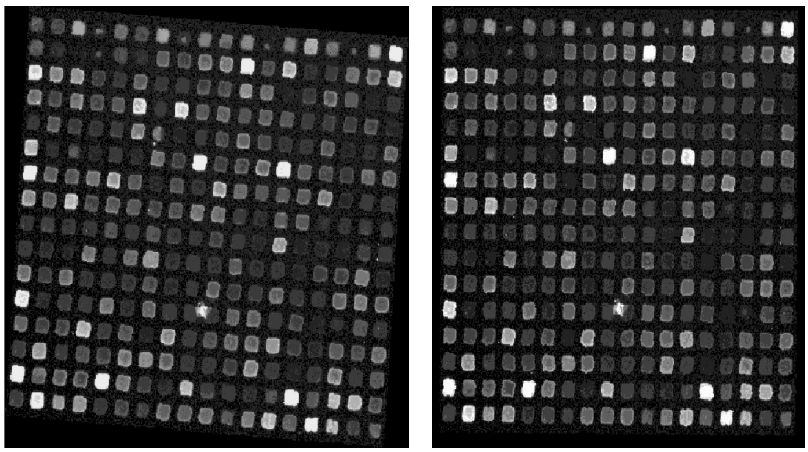

Fig 7 and 8: Clockwise skewed subgrid and Skew corrected subgrid, Image ID: 75757.

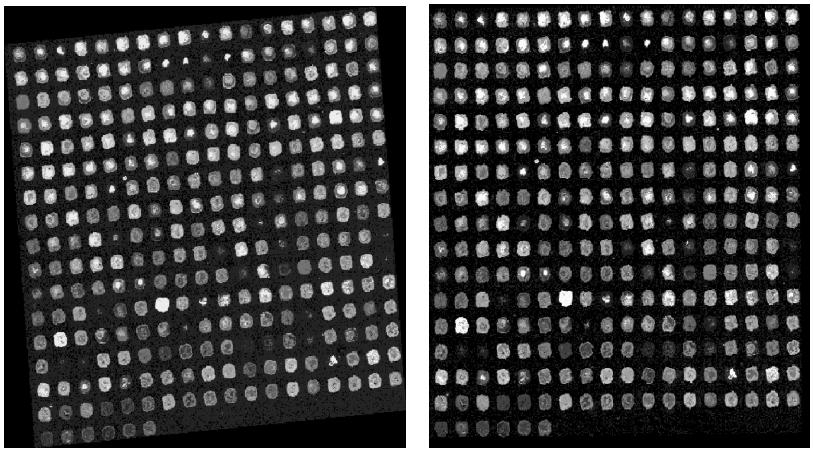

Fig 9 and 10: Anticlockwise skewed subgrid and Skew corrected subgrid, Image ID: 39114.

Table 1: estimated angle (φ) and execution time (τ_s) of the proposed skew correction algorithm.

Image ID	**Estimated angle in degrees (α)**	**Execution time in seconds (τ_s)**
75757 (TBDB)	3.7	20
42157 (Stanford)	2.8	15
37993 (UNC)	3.5	18
32045 (TBDB)	4.2	22

Adaptive Threshold and Filtering

Filtering is performed in 2 steps which are described in the 2 subsections below. Thresholds on spot size are first computed on segments of the image. Insignificant spots are filtered using these thresholds.

The skew corrected binary image is divided into n segments. Number of segments can be increased depending on the level of noise. The subgrid is divided into 4 segments in the proposed approach as follows

1st segment	2nd segment
Rows=0 to r/2	Rows= 0 to r/2
Columns=0 to c/2	Columns= c/2+1 to c
3rd segment	4th segment
Rows=r/2+1 to r	Rows= r/2 +1to r
Columns= 0 to c/2	Columns= c/2+1 to c

where r is the number of rows and c is number of columns of skew corrected image.

For each segment, the numbers of connected components are computed. The thresholds on spot size in each segment are calculated using the equation 3.1.

$$T(i) = \frac{\text{(Number of pixels in } i^{th} \text{ segment)}}{\text{(Total number of connected Components)}} \quad (3.1)$$

where i ranges from 1 to 4.

For example in the image (ID: 32040) Fig. 5 the number of bright pixels in the four segments are 5523, 5090, 6075, 2031. Total number of connected components is 894. The thresholds are 5523/894=6, 5090/894= 6, 6075/894=7, 2031/894=2

The results of the proposed filtering process in removing the insignificant spots using the threshold value and execution time (τ_f) are reported in Table 2.

Table 2: Estimated threshold value (A_T) and Execution time (τ_f) of the proposed filtering process.

Noisy Image ID	**Thresholds on a subarray**	**#spots in the subarray**	**Execution time(τ_f) in seconds**
35964 (Stanford)	T1=9 T2=12 T3=11	257	8

	T4=13		
32042 (TBDB)	T1=6 T2=6 T3=7 T4=9	894	12
32070 (TBDB)	T1=13 T2=12 T3=12 T4=10	2350	20
42247 (TBDB)	T1=16 T2=16 T3=11 T4=11	780	10

Execution time for the filtering process is proportional to number of spots in a noisy microarray image. Adaptive thresholds obtained in the previous step are used to filter insignificant noisy spots in the segments. If the number of pixels in a component are less than threshold value ($_{T(i)}$) in each segment, then remove the spot (insignificant spot) by setting intensity zero to all pixels in that component. The idea behind using adaptive threshold is, if in a subarray should few successive columns or rows have tiny spots filtering using global threshold will eliminate all these spots. This results in sparse grid lines.

Shown in figure 11 is a noisy microarray image. Figure 12 is the noise free filtered image.

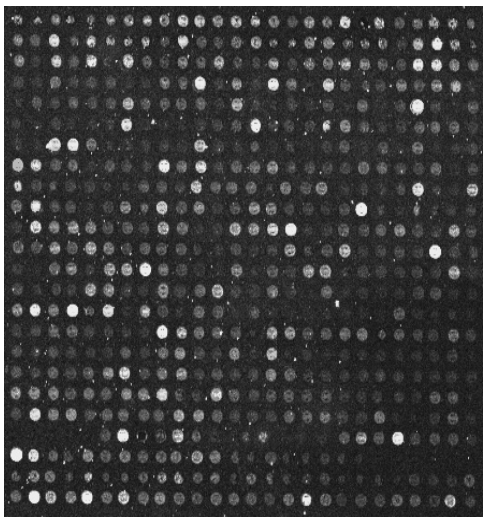

Figure 11: Noisy Microarray Image ID:32040.

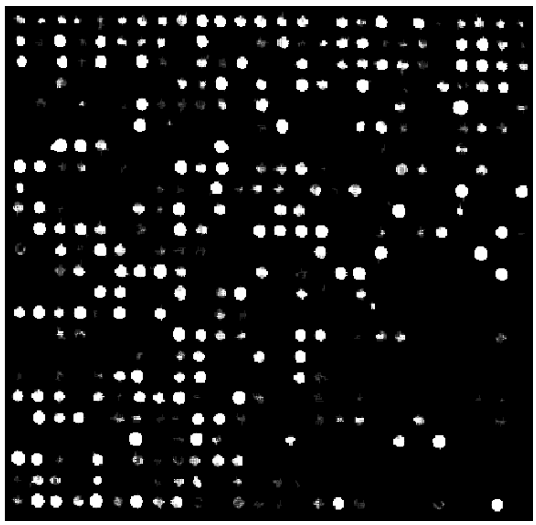

Figure 12: Filtered Microarray Image ID: 32040.

Automatic Gridding Process

Automatic gridding is performed in 3 steps which are described in the 3 subsections below.

Determination of position of grid lines

For each connected component in the filtered image, rmin, rmax, cmin, cmax are determined as shown in figure 13. Sorted arrays of rmin values (similarly rmax, cmin, cmax values) are found. Array of successive differences of rmin array called diff_rmin also for rmax, cmin, cmax (diff_rmax, diff_rmin, diff_cmin, diff_cmax) is found. Key portions of rmin, rmax and diff_rmin, diff_rmax are shown below. All computations done on image ID (62919).

rmin:

60	213	9	110	297	77	246	76	315	9	246
315	43	110	128	9	25	212	333	366	109	60
383	93	129	146	313	76	212	145	128	213	25
211	112	399	163	351	24	177	76	399	297	92.

rmax:

72	226	25	123	310	89	259	90	327	21	258
327	55	121	142	20	38	224	341	377	122	71
394	105	140	156	324	89	224	158	140	222	37
224	124	410	172	360	38	191	87	410	306	104

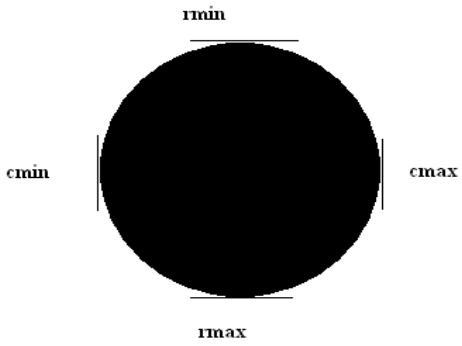

Figure 13: Representation of rmin, rmax, cmin & cmax of a spot.

The steps below describe determination of horizontal grid lines.

1. The rmin array is sorted in ascending order

sorted rmin:

9	9	9	24	25	25	43	60	60	76	76
76	77	92	93	109	110	110	112	128	128	129
145	146	163	177	211	212	212	213	213	246	246
297	297	313	315	315	333	351	366	383	399	399

2. The differences of successive rmin values in the sorted rmin array are calculated.

diffrmin:

0	0	15	1	0	18	17	0	16	0	0
1	15	1	16	1	0	2	16	0	1	16
1	17	14	34	1	0	1	0	33	0	51
0	16	2	0	18	18	15	17	16	0	

3. Sudden change in the difference in rmin values indicate the end of previous row of spots and beginning of next row of spots.

Observe the sudden change from 0 to 15, at position 3 in diff_rmin array. The third element of rmin array is 9. Hence examination of rmin diff_rmin suggests a grid line at row 9. Similarly it is understood that successive values of grid rmin.

grid rmin:

9	25	43	60	77	93	112	129	146	163	177
194	213	230	246	263	280	297	315	333	351	366
383	399									

Similarly grid_rmax is determined. Shown below are sorted_rmax, diff_rmax, grid_rmax values.

sorted rmax:

20	21	25	37	38	38	55	71	72	87	89
89	90	104	105	121	122	123	124	140	140	142
156	158	172	191	222	224	224	224	226	258	259
306	310	324	327	327	341	360	377	394	410	410

diff_rmax:

1	4	12	1	0	17	16	1	15	2	0	1
	14	1	16	1	1	1	16	0	2	14	2
	14	19	31	2	0	0	2	32	1	47	4
	14	3	0	14	19	17	17	16	0		

grid_rmax:

25	38	55	72	90	105	124	142	158	172	191
208	226	243	259	276	293	310	327	341	360	377
394										

Finally, positions of horizontal gridlines are determined by finding average of rows suggested by grid_rmin and grid_rmax contents. Thus horizontal gridlines are placed at rows 9, 25 (25+25/2), 41 (38+43/2), 57 (55+60/2)…etc.

In a similar manner vertical gridlines are positioned using sorted_cmin, diff_cmin, grid_cmin, sorted_cmax diff_cmax, grid_cmax.

Construction of rectangular grid

Consolidated rmin, rmax, cmin and cmax are used to build the grid structure. Figure 14 describes the coordinate system for the grid structure using above mentioned values. Point A coordinates (grid_rmin, grid_cmin) represents top left corner, B coordinates (grid_rmin, grid_cmax) represents top right corner, C coordinates (

grid_rmax, grid_cmax) represents bottom right corner and finally D coordinates (grid_rmax, grid_cmin) represents bottom left corner of the rectangular grid..

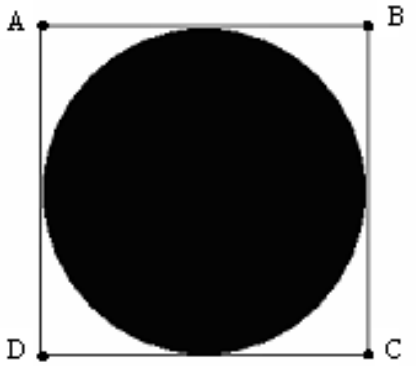

Figure 14: Coordinate system of the spot for rectangular grid.

Distance between each of the spot along the row and column is estimated to avoid the overlapping of the rectangular grid with the adjacent spot. The major advantage of the proposed approach is that the next stage of the microarry image analysis, segmentation can be performed easily and with minimum errors.

Experimental Results

In this section the performance of the proposed approach is evaluated on real noisy microarray images from SMD (Stanford microarray database), UNC (University of North California microarray database) and TB database. The images are available for free download from https://genome.unc.edu , http://www.tbdb.org/cgi-in/data/clickable.pl.html.

Figure 15. shows one subgrid of noisy microarray image. As discussed in section 3.2, Adaptive threshold is used to perform filtering. Figure 16. shows filtered image from which the observation shows that, most of the contaminated (insignificant, noisy) pixels are removed. Figure 17 shows noisy microarray image .Figure 18 filtered image using proposed approach. Figures 19, 20 and 21 shows the grid structure for the noisy microarray image using the proposed approach.

The execution time for the filtering process using adaptive threshold technique is proportional to number of spots in the noisy microarray image and the execution time for detecting and correcting the skew angle depends on, the number of spots in each subgrid.

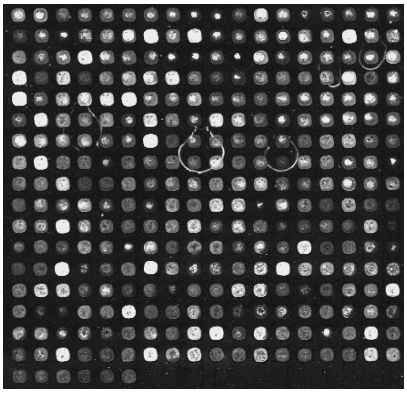

Figure 15: Subgrid of noisy microarray image, Image ID: 39119 , Database: TBDB.

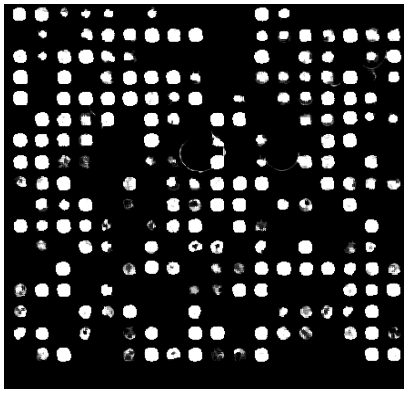

Figure 16: Filtered image using adaptive threshold, Image ID: 32919.

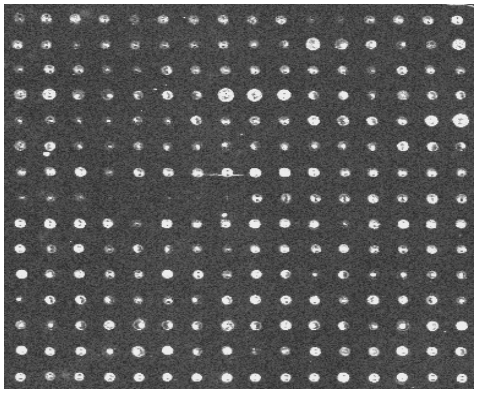

Figure 17: Subgrid of Noisy Microarray image. Image ID: 35964.

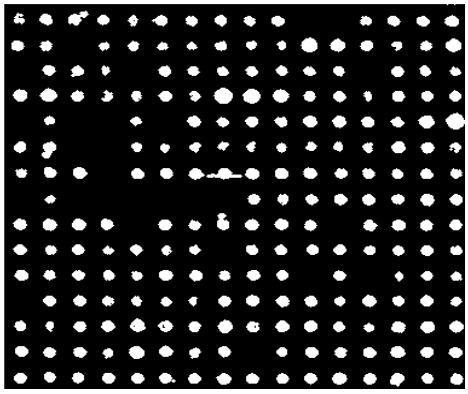

Figure 18: Filtered Image using adaptive threshold, Image ID: 35964.

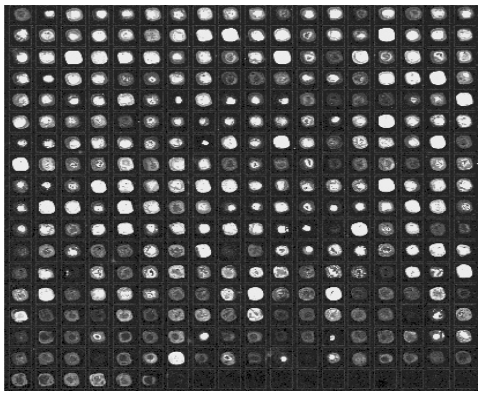

Figure 19: Grid of microarray image using our approach, Image ID: 38052 Database: TBDB.

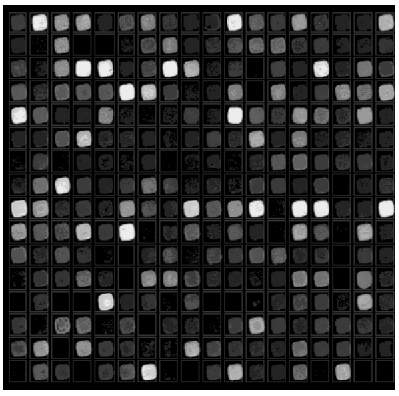

Figure 20: Grid of microarray image using our approach, Image ID: 75186 Database: TBDB.

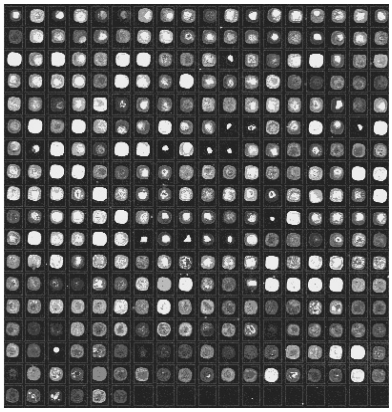

Figure 21: Grid of microarray image using our approach, Image ID: 37010 Database: TBDB.

Conclusion

In this paper, we have presented a new spatial topology technique to automatically grid skewed and noisy microarray images. Skew detection and correction is performed using corner spot detection approach. The noise removal is performed through adaptive thresholding, by which the entire process is robust, in the presence of noise, skew, artifacts and weakly expressed spots. Finally the gridding is performed based on spatial topology of spots. To summarize the three stages of the proposed approach when executed in sequence, it is effective and computationally simple.

References

[1] M. B. Eisen, ScanAlyze Nov. 1999 [Online]. Available: http://rana.lbl. gov/EisenSoftware.htm

[2] J. Buhler, T. Ideker, and D. Haynor, Dapple: Improved techniques for finding spots on DNA microarrays UW CSE Tech. Rep. UWTR 2000-08-05, Aug. 2000, pp. 1–12.

[3] Biodiscovery, Inc., ImaGene 2005 [Online]. Available: http://www. biodiscovery.com/imagene.asp

[4] P. Hegde *et al.*, "A concise guide to cdna microarray analysis," *Biotechniques*, vol. 29, no. 3, pp. 548–556, Sep. 2000.

[5] A. N. Jain, T. Tokuyasu, A. Snijders, R. Segraves, D. Albertso, and D. Pinkel, "Fully automatic quantification of microarray image data," *Genome Res.*, vol. 12, pp. 325–332, 2003.

[6] A. W. Liew, H. Yan, and M. Yang, "Robust adaptive spot Segmentation of DNA microarray images," *Pattern Recognit.*, vol. 36, pp. 1251–1254, 2003.

[7] K. Hartelius and J. M. Carstensen, "Bayesian grid matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 162–173, Feb. 2003.

[8] M. Katzer, F. Kummert, and G. Sagerer, "A Markov random field model of microarray gridding," presented at the ACM Symp. Applied Computing 2003.

[9] N. Deng and H. Duan, "The automatic gridding algorithm based on Projection for microarray image," in *Proc. 2004 Int. Conf. Intell. Mechatronics Automation*, Chengdu, China, 2004, pp. 254–257.

[10] A. W. Liew, H. Yan, and M. Yang, "Robust adaptive spot Segmentation of DNA microarray images," *Pattern Recognit.*, vol. 36, pp. 1251–1254, 2003.

[11] L. Rueda and V.Vidyadharan, "A hill-climbing approach for automatic gridding of cdna microarray images," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 3, no. 1, pp. 72–83, Jan.-Mar. 2006.

[12] G. Antoniol and M. Ceccarelli, "A markov random field approach to microarray image gridding," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Cambridge, U.K., 2004, pp. 550–553.

[13] H. Y. Jung and H. G. Cho, "An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis," *Bioinformatics*, vol. 18, no. 1, pp. 141–151, Oct. 2002.

[14] P. Bajcsy, "Gridline: Automatic grid alignment in DNA microarray scans," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 15–25, Jan. 2004.

[15] M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof, and J. O'Brien, "Automated image analysis for array hybridization experiments," *Bioinformatics*, vol. 17, no. 7, pp. 634–641, Jul. 1, 2001.

[16] N. Brandle, H. Bischof, and H. Lapp, "Robust DNA microarray image analysis," *Mach. Vis. Appl.*, vol. 15, pp. 11–28, 2003.

[17] J. Ho, W. L. Hwang, H. H. S. Lu, and D. T. Lee, "Gridding spot centers of smoothly distorted microarray images," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 342–353, Feb. 2006.

[18] G. Antoniol and M. Ceccarelli, "Microarray image gridding with stochasticsearch based approaches," *Image Vison Comput.*, vol. 25, no. 2, pp. 155–163, Feb. 2007.