Title: Preprocessing for DNA Images

Authors: Lili Ma and YangQuan Chen

Center for Self-Organizing and Intelligent Systems (CSOIS)
Department of Electrical and Computer Engineering
College of Engineering, Utah State University
4160 Old Main Hill, Logan Utah 84322-4160, USA
Emails: lilima@cc.usu.edu, yqchen@ece.usu.edu
Web: http://www.csois.usu.edu/

Date: First draft in August 2003, Revised in December 2004.

Technical Report No. : USU-CSOIS-TR-04-07

Reports are available from http://mechatronics.ece.usu.edu/reports/

Preprocessing of DNA Arrays

Lili Ma and YangQuan Chen
CSOIS, ECE Dept., Utah State University
lilima@cc.usu.edu, yqchen@ece.usu.edu

1. Introduction

The objective of the microarray image analysis is to extract probe intensities or ratios at each cDNA target location, and then cross-link printed clone target information so that biologists can easily interpret the outcomes and perform further high-level analysis. Preprocessing of DNA arrays have been a hot research field these days. In this work, we report our efforts in this area.

2. Image Processing

The objective of the image processing module is to extract the features of DNA arrays, more specifically, the centers and the radii of the circles representing the DNA patterns and other related information as required, given an image containing certain DNA pattern and some known knowledge of the number of cells in the image. In our paper, we know beforehand that the input image contains 8×12 cells of equal size. Figure 1 shows the detailed procedures of feature extractions, where the global segmentation, edge detection, and circle fit are described in the following:

(1) Global segmentation: segmentation of the image for each cell based on known information of the number of cells contained in the image.
(2) Edge detection: extraction of each cell boundary.
(3) Circle fit to edge pixels: circle fit to the extracted cell boundary.

Figure 1: Flow chart of the image processing module.

Global segmentation is first performed for an input image. After the global segmentation, 8×12 subimages are extracted and each subimage is processed to output the center and radius of a circle. Detailed global segmentation and circle fitting algorithms will be described in the next. The outputs will possibly include the density information and the variance of the center & radius in the future.

2.1 Global Segmentation

The objective of the global segmentation is to locate the regions of each subimage or cell. We applied some known knowledge to the input image, i.e., the knowledge that there are 8×12 cells in the image. This known knowledge helps to extract the exact locations of the cells. Here, we use the term “global” to emphasize that the segmentation, or partition, is achieved after a global balance of the input image. Without this step and just by applying a universal threshold to the whole image, cells of insignificant densities might not be detected successfully. After the global segmentation, each subimage is processed locally using a local threshold for its edge detection. Some previously unobserved densities when applying a universal threshold to the whole image can be recognized.

Figure 2 illustrates the detailed steps of the global segmentation algorithm, which are described below:
(1) Input gray-level image.
(2) Binary image after applying a threshold to the input image where the threshold is determined using the well-known Otsu's method **[1]**.
(3) Coarsely extracted edges of the cells in the first two leftmost columns. Circle fit to the detected cell boundaries. The output of the circle fit is the center and radius of the detected edge boundary.
(4) Line fit to the centers of the fitted circles. Computation of the distance between the two fitted lines.
(5) Partition of the whole image into individual cells, using the angles of the two fitted lines, the distance between the two fitted lines, and the knowledge of totally 8×12 cells of equal size.

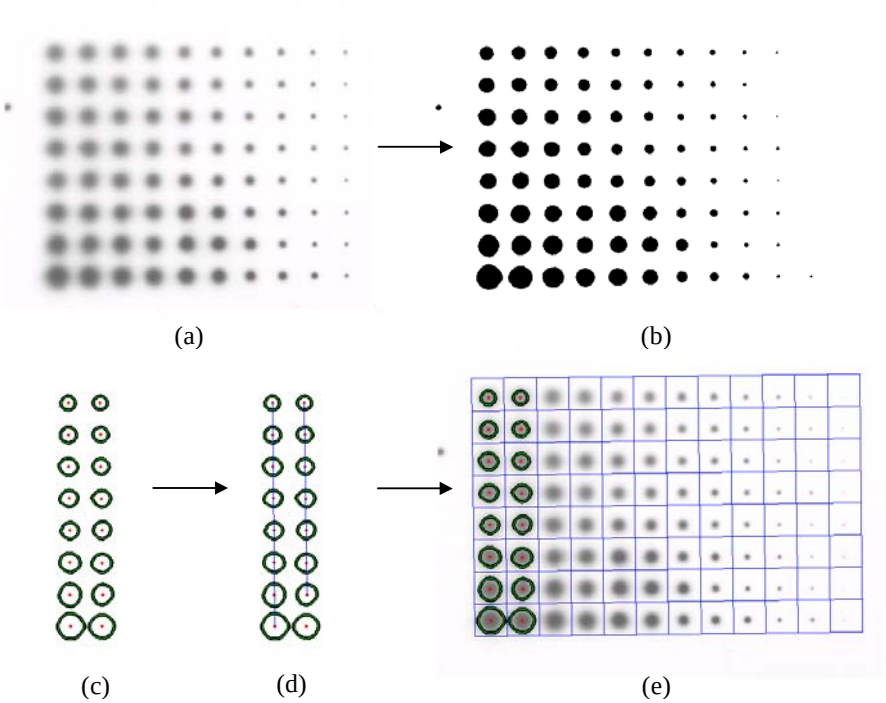


Figure 2: Global segmentation of the input image.

2.2 Circle Fit

Given the extracted boundary of a cell, we want information such as the center and the radius of the detected DNA pattern. When the radius of a circle is unknown, a simple algebraic circle fitting method can be applied for this purpose in the least square (LS) sense. That is, given a set of points (edge pixels) with coordinates $\{(x_i, y_i), i = 1,2,\ldots,n\}$, find the best parameters $(a_1,\ldots,a_4)$ in the circle equation [2]

$$a_1(x^2 + y^2) + a_2 x + a_3 y + a_4 = 0,$$

such that the algebraic error is minimized. Denote $a = [a_1, a_2, a_3, a_4]^T$ and construct a matrix D as:

$$D = \begin{bmatrix} {x_1}^2 + {y_1}^2 & x_1 & y_1 & 1 \\ {x_2}^2 + {y_2}^2 & x_2 & y_2 & 1 \\ {x_3}^2 + {y_3}^2 & x_3 & y_3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ {x_n}^2 + {y_n}^2 & x_n & y_n & 1 \end{bmatrix}.$$

We can have

$$Da = 0.$$

The circle parameters a can be obtained as the least square solution of the above equation. A typical circle fit to the extracted cell boundary is shown in Fig. 3.

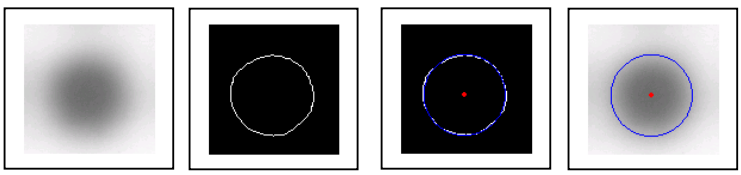


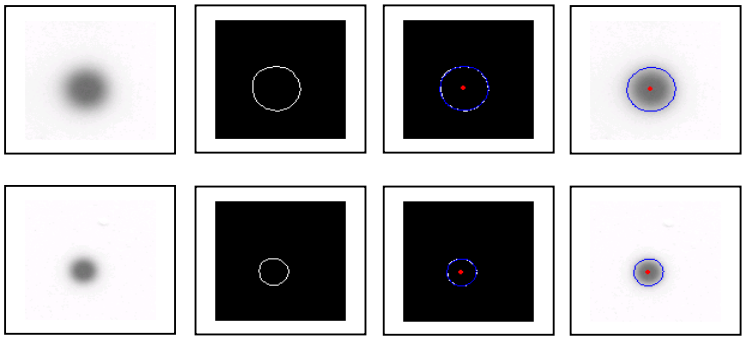
Figure 3: Algebraic circle fit to the cell boundary.

2.3 Boundary Extraction of Each Subimage

Given the subimages segmented using the global segmentation method described earlier, three functions are applied to extract the boundaries of the cells, where the second function is applied when the first function fails, and similarly, the third function is applied when the second fails. The three functions act in a sequence based on the number of the detected edge pixels of each subimage.

Function 1:

When the number of the detected edge pixels of a subimage is greater than a predefined threshold, *threshold1* (threshold1 equals to 25 in our current implementation), the edge pixels are fed into the algebraic circle fit function described before, whose outputs are the center and the radius of the edge pixels. Examples of the patterns that can be recognized by Function 1 are shown in Fig. 4.

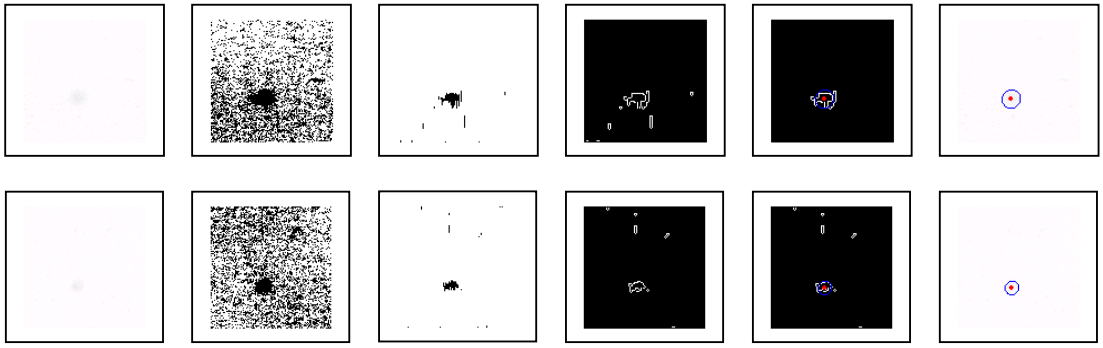


Input subimage → after edge detection → circle fit to the edge pixels → fitted circle in the original subimage.

Figure 4: Illustration of the circle fit when applying Function 1.

Function 2:

When the observed cell in the subimage is small, or the density is low, or the detected edge pixels are noisy, Function 1 may fail. In this case, image enhancement operators need to be applied for noisy pixel removal. Simple image dilation operator to remove small objects is used for this purpose **[3] [4]**. After the image dilation, when the number of pixels in the detected edge is greater than another predefined threshold, *threshold2* (threshold2 equals 20 in our current implementation), these edge pixels are fed into the circle fit function to give the center and radius. Examples of the cells that can be recognized by Function 2 are shown in Fig. 5. In our implementation, the predefined thresholds, the *threshold1* and *threshold2*, are manually tuned.

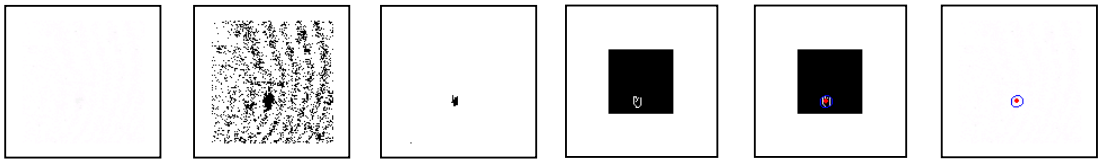


Input subimage → after edge detection → after image dilation operation
→ edge detection again → select the largest group of pixels → circle fit in the original subimage.

Figure 5: Illustration of the circle fit when applying Function 2.

Function 3:

When the observed cell in the input subimage is extremely small, Function 2 may fail. Since the image dilation operation performed in Function 2 has an adverse effect of making the objects smaller, in Function 3, we search through a region close to the center for connected components of largest size. Doing this way, we assume that the DNA pattern, if any, evolves around the center of each cell. Examples of the cells that can be recognized by Function 3 are shown in Fig. 6.



Input subimage → after edge detection → after image dilation operation → edge detection of a region close to the center → search for connected components of largest size → circle fit in the original subimage.

Figure 6: Illustration of the circle fit when applying Function 3.

It needs to be pointed out that with the above three functions, it is not guaranteed that we can always detect a circle for a given subimage. However, it is noticed that the situations where the above three functions fail is when the pattern is not even recognizable by our human eyes. In these cases, there is barely anything noticeable in the given subimage.

2.4 Final Processed Images

Using the global segmentation and the three functions described before, Fig. 6 shows three examples of the processed images where the fitted circle are displayed in the original input images. Notice that, in the first two leftmost columns, circles that are plotted in green denote those fittings used for global segmentation. After the global segmentation, local thresholds are used for the detection of the edge pixels of individual cells. Circle fittings to the locally detected edge pixels are plotted in blue.

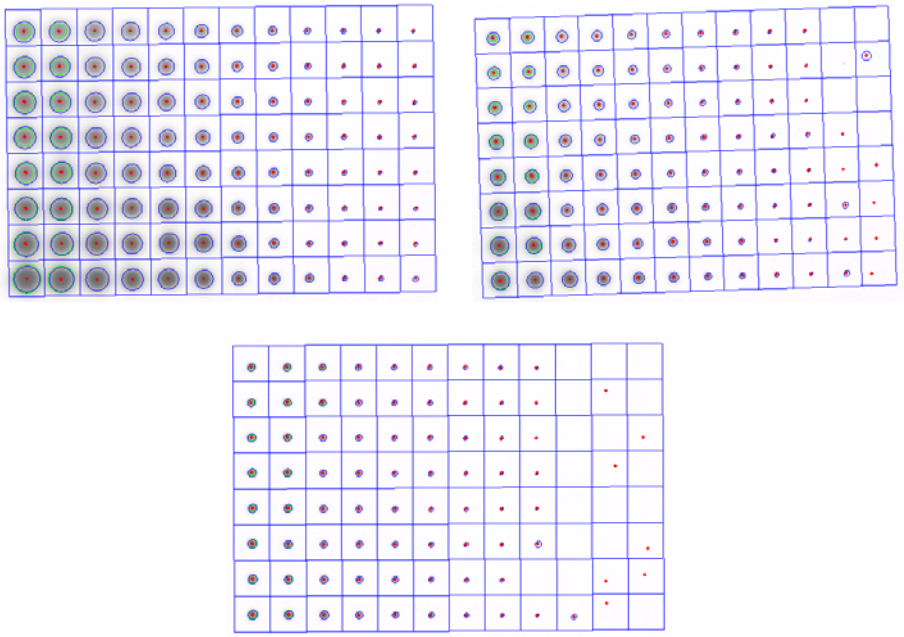


Figure 7: Examples of final processed images.

3. Discussions and Future Development

The procedures given in this work represent our start-up development for DNA array processings. Future works will be directed to achieve more efficient, robust, and accurate processing performance. Improvement can be expected by investigating the following issues:

(1) Comparison of edge detection algorithms. Candidates include:
- Edge detectors provided as Matlab routines, such as the Sobel, Prewitt, Roberts, and Canny detectors [3].
- The SUSAN edge detector **[5] [6]**.
- Fractional-order edge detectors **[7]**.

(2) Fine adjustments of the global segmentation for images tilted with significant angles.
(3) Further investigations on the tuning knobs, the predefined thresholds.
(4) Processing of multiple images that are a function of time.

4. Acknowledgement

The authors would like to thank Dr. Dong Chen for providing the DNA images.

5. References

[1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. on Systems Man Cybernetics*, 9 (1), 62-69, 1979.
[2] Zhen Song, Yangquan Chen, Lili Ma, and You Chung Chung, "Some Sensing and Perception Techniques for an Omnidirectional Ground Vehicle with a Laser Scanner," *IEEE Int. Symposium on Intelligent Control*, 690-695, October 2002.
[3] David Forsyth and Jean Ponce, "Computer Vision - A modern approach," Third Edition, Prentice Hall, August 2003.
[4] Matlab Image Processing Toolbox Manual.
[5] S. M. Smith and J. M. Brady, "SUSAN - A New Approach to Low Level Image Processing," Technical Report, http://vis-www.cs.umass.edu/~riseman/susan.pdf.
[6] S. M. Smith and J. M. Brady, "SUSAN – A New Approach to Low Level Image processing," *Int. Journal of Computer Vision*, 23(1): 45-78, May 1997.
[7] B. Mathieu, P. Melchior, A. Oustaloup, and Ch. Ceyral, "Fractional Differentiation for Edge Detection," *Signal Processing*, 83(11), November 2003, 2421-2432.